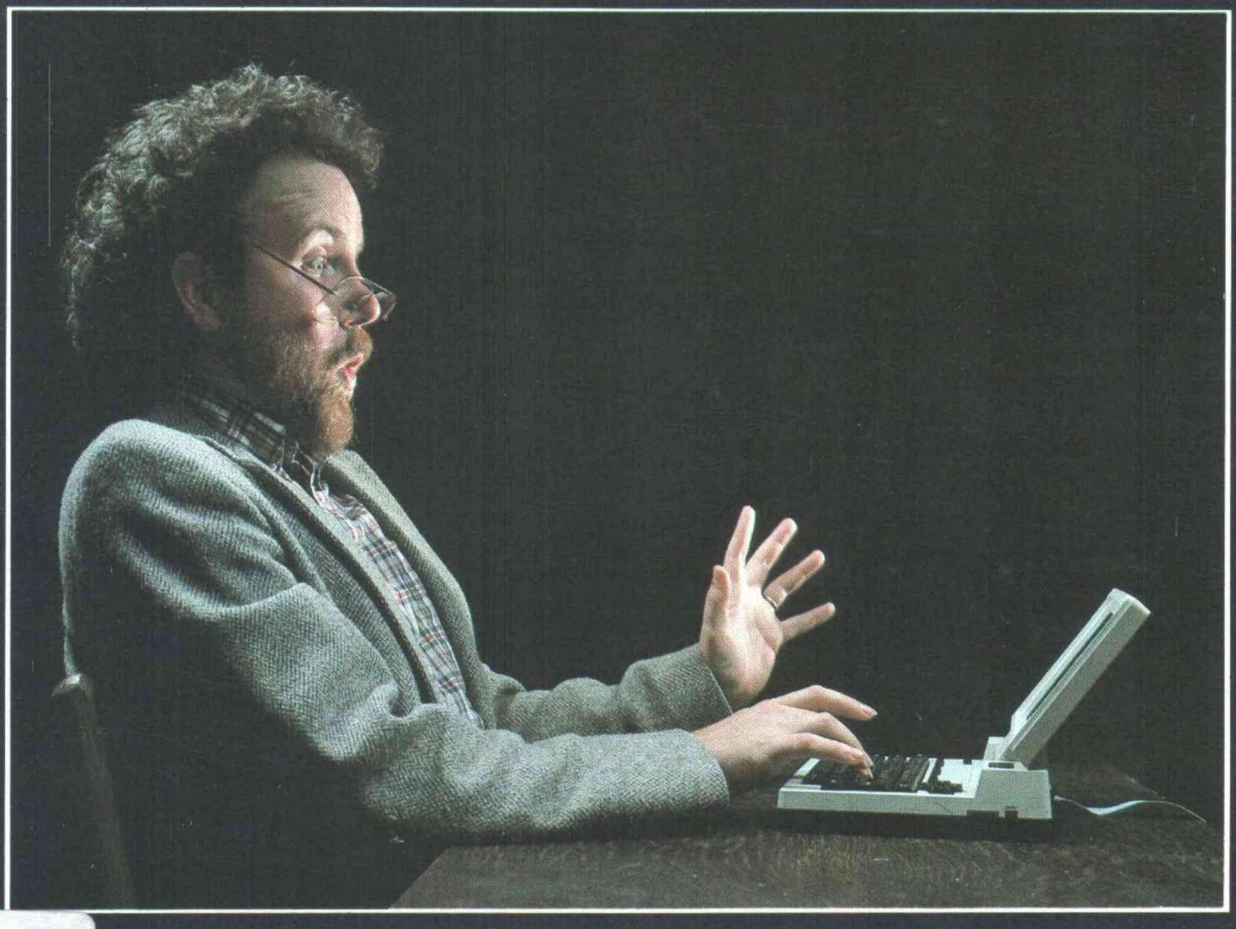
OCTOBER 1985 $3.95
VOLUME 3 ISSUE 2

portable 100/200

FOR USERS OF TANDY BRIEFCASE COMPUTERS



ULTIMATE ROM
IT WILL BLOW YOU AWAY

AN ORCHESTRA IN YOUR 100

Making Beautiful Music With Your Portable

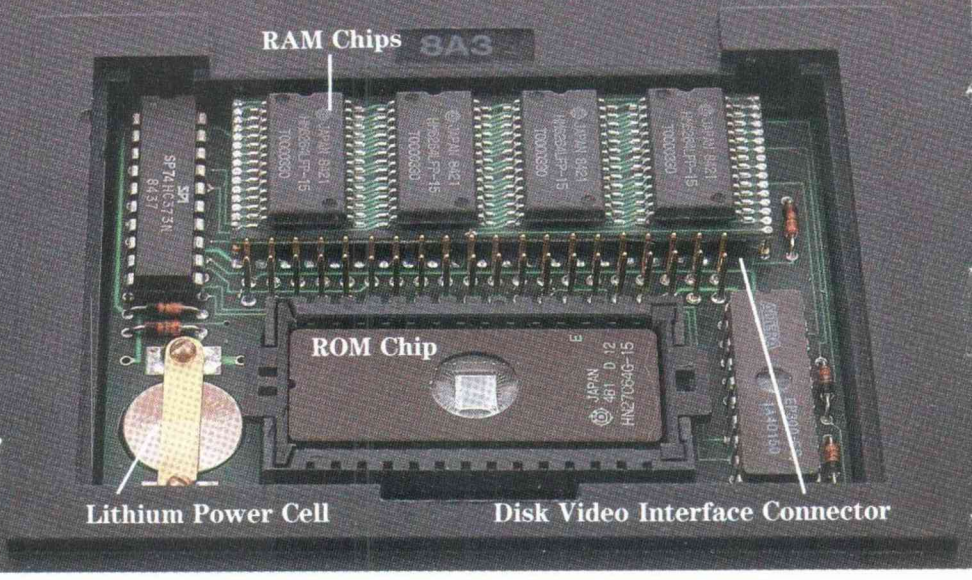
TURBO-BASIC

Building Programs For Speed

NULL-MODEM

A Do-It-Yourself Tip

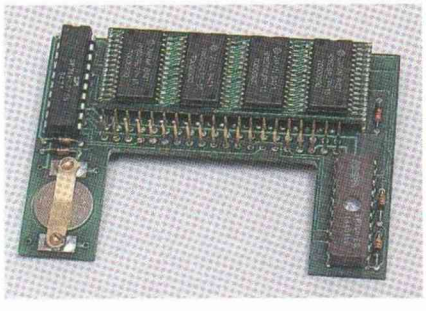
State of The Art RAM Technology



At last, a 96K Model 100!
"The new PG Design RAM should be in every Model 100 built!"

Miniaturized Technology

Our miniaturized RAM chips are state of the art. They are the most advanced memory chips found anywhere in the world. Their tiny size allows us to keep a low profile in the expansion port of the Model 100. We use a technique called *vapor phase soldering* to ensure that each and every tiny connection is clean — perfect.



The RAM module is precision constructed.

Like The Original

Each 32K RAM bank has its own command of the software that comes in your Model 100. BASIC, TEXT, TELCOM, ADDRSS, SCHEDL are all there in each bank and you can use them as you would in the original bank. Each bank can be accessed from any of the other banks. We even have an optional data transfer program which allows data to be transferred from one bank to any other bank. It's flawless!

Simple Installation

"Adding this 64K RAM module to your Model 100 is as easy as putting in new batteries." Once you've removed the expansion cover on the back of your Model 100, just snap the **PG Design** RAM module in. You can't get it wrong! The pins line up perfectly with the expansion holes in the Model 100 compartment. Snap the cover back on and turn your Model 100 over. Turn it on and enter BASIC. Type in the one line program we supply you and presto—you've got a Model 100 with 96K of RAM. You do not need a 32K Model 100 to utilize the **PG Design** 64K RAM module.

No Need To Remove It

The beauty of this RAM module is that we've enabled you to have a Model 100 with 96K of RAM and we've given you access to the other Model 100 options within the expansion compartment. The DVI connection can be made easily with our rugged connectors. Gone are the flat flimsy pins. And best of all, the ROM slot is clear to insert any ROM modules, (like Tandy's Multiplan on ROM). We designed this RAM module so it wouldn't ever have to be removed from your Model 100. But, if you should remove it, we've installed a lithium power cell that will keep all the data on the module intact for six months outside the Model 100. Six months! The actual life span of the lithium power cell while in the Model 100 is nearly six years!

Guarantee

We stand behind all the products we manufacture at **PG Design.** If you are not completely satisfied with your purchase, call us! If we cannot solve your problem, return the product to us and we will refund your money. We are positive that you will be completely satisfied with all our products.

Order Today

64K RAM module — $375
32K RAM module — $250
If you want only a 32K version of the RAM module, you may upgrade later for only $150. The 32K version is constructed exactly as the 64K module.

Data Transfer Program — call

Call us at 313/727-2744, or write. We accept Visa and MasterCard, as well as check or money orders. We ship within five days of receiving your order.

Dealer Prices Available

TRS-80 is a trade mark of Radio Shack

PG DESIGN ELECTRONICS, INC. *Micro-Computer Peripherals*

66040 Gratiot, Richmond, Mich. 48062 (313) 727-2744

Circle 39 on Reader Service Card

6 ROM BANK

plus powerful Nicad battery pack that gives 30 hours of power to your Model 100

$269

A joint PCSG/CRYPTRONICS project.

Now you can access LUCID, DISK+, WRITE ROM, RAM+ and others instantly.

The 6 ROM BANK plus battery pack lets you have the ultimate Model 100 system. When you get your unit just press in up to 6 ROMS into the sockets then push in a simple plug into your Model 100's ROM socket. The ROM bank has two posts on either end that insert into the two little holes on the underside of your Model 100.

The ROM bank props up the Model 100 at the same angle and height as those little legs you've seen. The ROM BANK itself is only about 1½" deep and it runs the width of your Model 100. It only weighs one pound. It not only installs instantly, but it pops free in a second if you need everything to lie flat in a briefcase.

Change from ROM to ROM with the touch of a thumb switch.

You can go from LUCID to WRITE to DISK+ to any other ROMS just by turning the thumb switch at the side of the ROM bank. The 6 ROM BANK is a sturdy well built construction that looks like it is a part of your Model 100.

What is also fantastic is that the ROM bank has a powerful NICAD battery and recharger built right in. This power source supplies up to 30 hours of life to your Model 100 with just 6 hours of recharge. What's nice is that it recharges right from your Model 100's power adapter. This is a quick charge system and if you need power in a hurry, you can get 6 hours of life for your Model 100 by just charging for an hour and half.

Suddenly, the Model 100 is a very powerful computer.

If you have the HOLMES portable disk drive with its powerful bundled database software package or the CRYPTRONICS 128K RAM expansion along with LUCID, WRITE ROM and DISK+ in the 6 ROM bank with its rapid rechargable NICAD power source, you have the ultimate portable system. It's all available only from PCSG.

As usual we don't want you to take our word for it. The 6 ROM BANK is sold on a 30 day trial. If you are not satisfied simply return within thirty days for a full refund. MC, VISA, AM. EX. or C.O.D.

1-214-351-0564

PORTABLE COMPUTER SUPPORT GROUP

11035 Harry Hines Blvd. No. 207, Dallas, Tx. 75229

VISA / MASTER CARD / AM. EXP. / C.O.D.

Circle 32 on Reader Service Card

portable 100/200

OCTOBER 1985 VOLUME THREE, NUMBER TWO

ON THE COVER

ARTICLES

REVIEWS

DEPARTMENTS

portable 100/200

Editor
J.D. Hildebrand
Managing Editor
Edward M. Jackson
Senior Editor
Park M. Morrison
Contributing Editor
Alan L. Zeichick

Production Director
Peter D. Koons
Art Director
Paige Garland Parker
Staff Artists
Elizabeth Maritato
Marjorie Strauss
Judith E.C. Webster

Advertising Director
David Hollis
Circulation Director
Nancy A. Wight
Circulation Manager
Loriel Peters
Subscriber Relations
Melissa R. Andrews
Mary M. Wight
Business Manager
April A. Jenkins
MIS Director
Bonnie Hellevig

Publisher
James S. Povec
Associate Publisher
Carl Cramer
Editorial and Business Offices:
Camden Communications, Inc.
P.O. Box 250
Highland Mill
Camden, ME 04843
(207) 236-4365
CompuServe ID: 76703,372
Fax Tel. (207) 236-4738

PORTABLE 100/200 is published monthly by Camden Communications, Inc.: James S. Povec, president; Carl Cramer, vice president. Application for mailing at second class rates pending at Camden, ME and additional mailing offices. Postmaster: Address changes to Camden Communications, Inc., P.O. Box 250, Camden, ME 04843.

Contributors: Manuscripts and photographs are welcomed. Publisher assumes no responsibility for the return of unsolicited manuscripts, queries or artwork. Materials submitted for consideration should be accompanied by a self-addressed, stamped envelope.

Subscriptions are $32 for 12 issues. Users group subscription rates available. Please inquire for delivery rates outside the U.S. Dealer inquiries invited.

ISSN 0738-7016

THINGS ULTIMATE

The letters are just starting to come in about the column that filled this space in August. In that column, *The Waiting ROM,* we attempted to put readers in our place and asked them to grapple (as we have grappled) with the problem of announced — but unavailable — products.

Our conclusion, which we put forth as a tentative position statement, was a compromise. We said we'd run product announcements when products are shipped and news reports when they're merely announced. In reviews we'll make judgments about how ready products are for release. And we won't attempt to censor ads — the advertisers buy that space, and it's theirs to use as they please.

At least one reader has suggested that when we allow premature advertising to appear in the magazine we're accomplices to fraud. A software vendor whose company has, to our knowledge, never advertised products before they were ready called to complain. "You can't be serious," he said.

Well, we *are* serious.

Today is August 20. All the advertising materials for the October issue must be at our offices by the end of the week to give us enough time to put the magazine through production and mail it to readers. This issue may contain ads for products that vendors plan to introduce September 15. Shall we turn them down because the products aren't yet available? What if, due to unforeseen technical difficulties, the introductions are delayed until November 15? Are we then accomplices to fraud?

Suppose we learn that a product advertised in our June issue wasn't delivered to customers until August. Do we write an article advising readers of the problem and suggesting that they don't deal with the vendor? Haven't we trespassed beyond the reasonable bounds of editorial coverage then, and begun to diminish the vendor's ability to do business in the future?

Our editorial position is open to reassessment. We reassess it all the time. But so far we're not convinced *anyone* is served if we refuse to carry the vendor's message to our readers.

CHANGES IN PORTABLE-LAND

This month's *Portable 100/200* brings with it a change of editorial leadership and administration. *Bruce Taylor* and *Nancy Laite,* under whose capable co-direction the magazine has flourished for the past six months or so, have left the company to pursue new careers. Nancy's departure is particularly bittersweet; hired on as a typist when the magazine was founded, she rose through the ranks during the first two years until she became managing editor, responsible for determining, by and large, the direction of the publication.

A new team is in place. *Park M. Morrison,* an experienced and talented editor who has spent the past couple of years as a VAR serving the publishing industry, takes over as senior editor. With assistance from managing editor *Ed Jackson,* Park will oversee *Portable 100/200's* editorial component on a day-by-day basis.

I joined the staff six months ago as technical editor after a two-year stint with the competition: *Portable Computer,* a now-defunct monthly that sought to cover the handheld, laptop and transportable markets. I often envied *Portable 100/200's* editors their ability to concentrate in depth on one computer instead of offering token coverage to more than 100; now I'm in their shoes and I'm finding them a comfortably roomy fit.

PLENTY TO CHEW ON

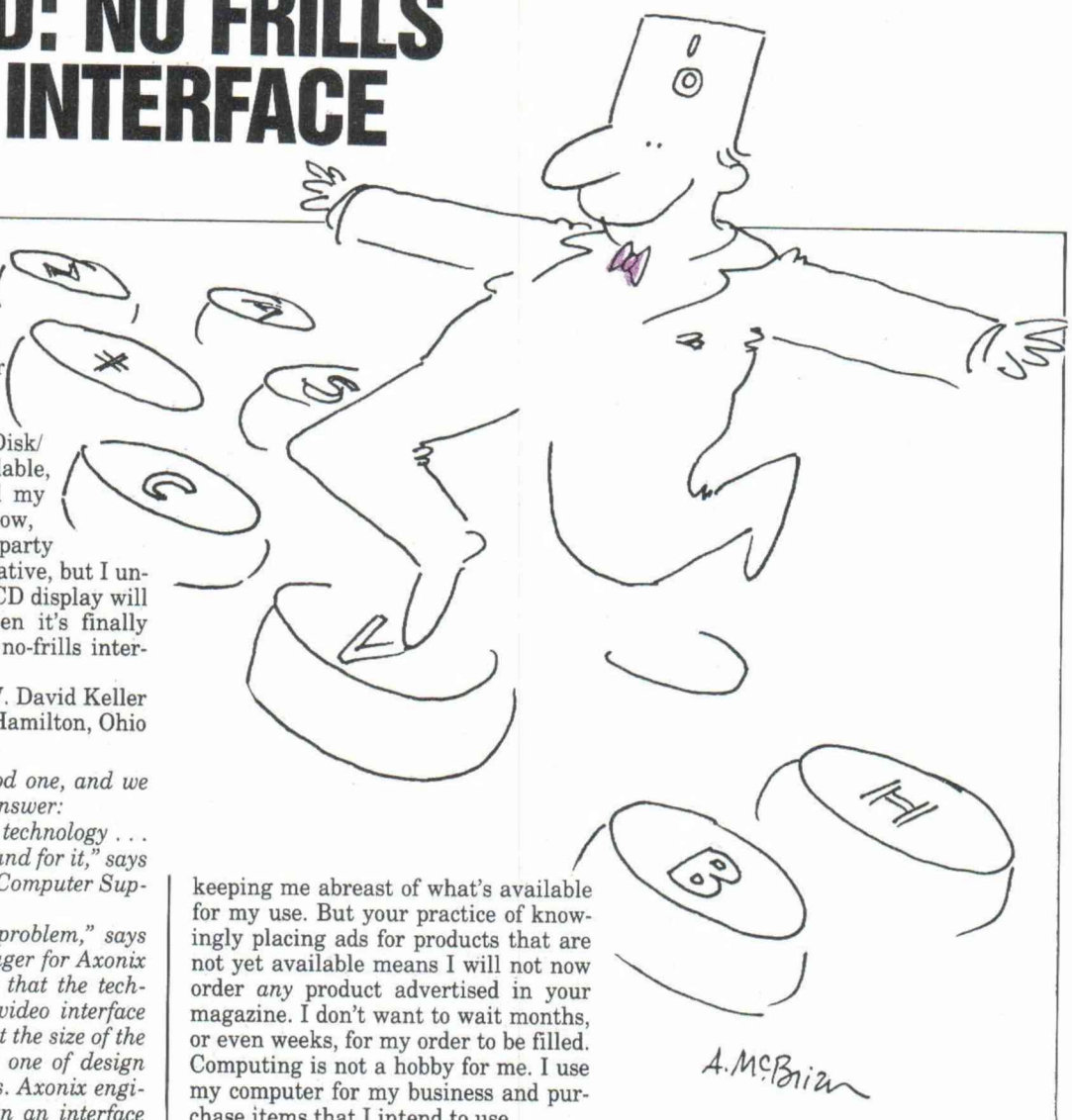
So there you have it. The new team's in place, fired up and ready to go. This issue is our response to everything we've learned about you from reading back issues, consulting with the former editorial team and analyzing the results of reader surveys.

We include a hardware construction project for those of you who feel comfortable with a soldering iron in hand. We've got plenty of programming articles, including novice-level tips for improving program performance, a nifty program lister that shows how the 100's operating system deals with binary program files and even a musical note taker for you frustrated musicians out there. Plus, we've reviewed a couple of the hottest products on the market: Traveling Software's Ultimate ROM and Axonix's Thinview.

It's a big, meaty package — just the way *Portable 100/200's* founding editors would have wanted us to start the third year of publication.

J.D. HILDEBRAND
EDITOR

WANTED: NO FRILLS VIDEO INTERFACE



Why hasn't Tandy or another vendor offered a reasonably priced video interface that provides a 40-character, 25-line display with a TV or an 80-character, 25-line display with a monitor?

I realize that Tandy's Disk/Video Interface is available, but at $799 it's beyond my reach. And as far as I know, Axonix is the only third-party vendor to offer an alternative, but I understand its Thinview LCD display will sell for about $500 when it's finally available. I'd jump at a no-frills interface in the $200 range.

W. David Keller
Hamilton, Ohio

Your question is a good one, and we called around to get an answer:

"We certainly have the technology . . . but there isn't much demand for it," says Mike Anders of Portable Computer Support Group.

"We're looking at the problem," says David Cox, general manager for Axonix Corporation, who agrees that the technology for a low-priced video interface exists, but disagrees about the size of the market. "The problem is one of design and marketing," Cox says. Axonix engineers are trying to design an interface that can be used with many different machines to make it more cost effective. Cox anticipates the company will offer an interface priced under $300 within three months.

Thanks for telling us — and them — what you want. —Ed.

ADVOCACY OR IMMORALITY?

I was very upset by your comments on advertisers in The Waiting ROM (August). I'm a daily user of my Model 100 and a subscriber to your magazine.

One of the most valuable parts of your publication is the advertisements. I'm not — and don't aspire to be — a programmer; so I buy and use a lot of dealers' software (and firmware and hardware).

I thought that your magazine was keeping me abreast of what's available for my use. But your practice of knowingly placing ads for products that are not yet available means I will not now order *any* product advertised in your magazine. I don't want to wait months, or even weeks, for my order to be filled. Computing is not a hobby for me. I use my computer for my business and purchase items that I intend to use.

I believe that advertising a product for sale when you're not prepared to fill orders constitutes criminal fraud. If you knowingly allow such advertising to appear in the magazine, I consider you an accomplice to the fraud. I will not renew my subscription when it expires if you do not change this policy.

Legitimate entrepreneurs don't finance their businesses by committing fraud. I certainly don't agree that you have to "fuel that entrepreneurial effort." On the contrary, I consider the practice unethical and immoral if not just plain illegal.

Dennis Loomis
Camino, California

ROM MUSINGS

I recently received a flyer from Portable Computer Support Group about Write ROM. It sounds like a very good product and I've almost decided to buy it.

However, I got to thinking. Lucid is also a very nice program, but buying both ROMs isn't just very expensive but would also be inconvenient. Every time I wanted to use a program I would have to remove one ROM and replace it with another. This is time-consuming, increases the chance of damaging the ROM and the computer, and would require me to carry one or both ROMs outside the computer.

We've done it again!!!

More super software for your Model 100, Tandy 200 & NEC PC8201A!

MEN-U-TILITY

Men-u-tility is a powerful new utility for your Model 100. Once installed it is completely automatic and comes up when ever you would normally return to the main menu. As you move the cursor bar over the files, the length of each file is instantly displayed in the upper right corner. Men-u-tility adds 8 function keys to your main menu. You can kill files, rename files, make files invisible, set the day, date and time without ever leaving the menu.

Men-u-tility is also a print formatter. With F3, you can print any .DO file to your printer and you decide the right and left margins, top and bottom margins and page length! F8 sets an alarm that will go off no matter what mode you are in, BASIC, TEXT, TELCOM. etc.

If you have the Disk Video Interface (not required), the menu will appear on whatever screen you are using.

Men-u-tility only requires 1.8K of RAM and won't conflict with your other machine language programs.(100)

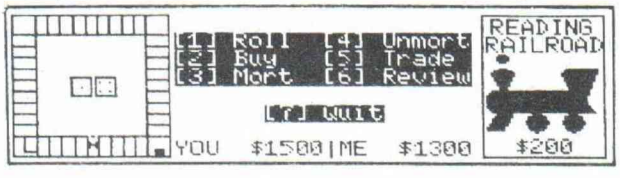
$24.95

LET'S PLAY MONOPOLY*

It's you against the computer and the computer is a tough competitor. The computer makes all its own decisions. Super fast machine language graphics display the whole board at all times. You can tell at a glance who owns what property and the number of houses on each. It never takes more than 1 or 2 seconds for the computer to decide what to do. The computer is such a good player that you'll be lucky if you even win half of the time. (100,200,NEC)

*Monopoly is a trademark of Parker Brothers.

$29.95



ASSEMBLER

Our assembler is the answer to your assembly language programming needs. It has all the features you expect in an assembler and more! It requires less than 3K of your valuable RAM space and is relocatable to any convenient place in memory. There are several useful macros already built in. You can output all or any portion of the assembled listing to your screen or printer. An extensive 56 page manual covers the use of the assembler, the complete 8085 instruction set, useful sample programs and LOTS of information on the ROM and reserved RAM areas. (100,200,NEC)

$32.95

Melody Maker

Melody Maker is a musical program generator. Simple cursor controls are used to select a note and position it on the staff making it easy to enter in sheet music. You can even use Melody Maker to add musical routines to your own programs. (100,200,NEC)

$19.95

BYTEFYTER

Now you can expand the memory capacity of your portable computer by reducing the size of the programs that you store in it. Bytefyter is a 100% machine language program that does just that. It is relocatable so that it won't conflict with any other machine language programs that you use now, or may use in the future.

Bytefyter works on your BASIC programs just as they are, IN PLACE. It strips unneeded spaces and remark lines. But that's not all! Bytefyter is smart! Bytefyter combines the lines of the BASIC program to whatever maximum length you specify. Each line of a BASIC program takes 5 bytes just for the line number and pointer information. By combining lines, Bytefyter saves a tremendous amount of space, space that could be used for another program or text file. Bytefyter actually checks the logic of your programs and doesn't combine lines that would cause the program to crash.

Bytefyter is amazingly fast. It will do its job on even the largest BASIC program in just seconds You'll want to use Bytefyter on all your BASIC programs, whether you wrote them or bought them. (100,200,NEC)

$24.95

RENUMBER

Renumber is a machine language program that lets you renumber the lines of your BASIC programs IN PLACE! Renumber adjusts all references to line numbers throughout the program. It is completely relocatable so it won't conflict with your other machine language programs.

Renumber is FAST! It will renumber even the largest BASIC program in just seconds. You can renumber all or just part of a program. You decide the starting line number and the increment to use. It couldn't be any simpler. This is one utility that the serious BASIC programmer just can't afford to be without! (100,200)

$24.95

CBUG

CBUG is the ultimate debugging tool for your lap computer. It only requires 3K of your precious RAM space and is relocatable to any convenient place in memory. CBUG is not just fast, small and easy to use, it is POWERFUL!! With CBUG you can step through an assembly language program or the ROM while it displays the registers, the status of the flags, and associated memory locations. You can set breakpoints and excecute your code to that point. You can step through call instructions with a single keystroke and return to the point after the call. CBUG does number base conversion, hex addition and subtraction, search and display, search and replace and block moves of memory. CBUG allows you to alter the values contained in the registers, display memory and loadvalues into memory like a monitor program. (100,200,NEC)

$29.95

SORT

Our Sort utility lets you sort any TEXT file in place. You can sort the file by any field. Sort is 100% machine language and only requires .8K of RAM. (100,200,NEC)

$19.95

CUSTOM SOFTWARE

1308 WESTERN ● WELLINGTON, KS 67152

316-326-6197

VISA MasterCard

Please include $1.00 per program for postage.
Be sure to specify Model 100, Tandy 200 or NEC PC-8201A.
For orders outside the U.S., include sufficient postage for airmail delivery. U.S. funds only.

Why couldn't both programs be put on one ROM? Perhaps I'm displaying my ignorance, but they're supposed to be 32K ROMs, aren't they? It's hard for me to believe either Write ROM or Lucid is over 16K. Another nice thing to include on the same ROM would be Traveling Software's T-View 80. Would it be possible to do this?

Along the same line of thought, why couldn't someone program a custom operating system on a ROM to replace the built-in ROM supplied with the machine. I know it probably would be a lot of work, but it's possible, isn't it?

Mike Mahaffey
New York, New York

You've hit on some of the most pressing issues facing laptop computers today. Let's take your questions one at a time.

First, it isn't possible to put both Write ROM and Lucid on one 32K ROM. According to Portable Computer Support Group, each of the programs stretches a ROM's 32K capacity to the limit. PCSG has, however, begun advertising an external ROM bank that allows you to plug in as many as six ROMs at once and switch between them at will. We haven't tried the unit yet, but it sounds like a good idea.

T-View 80, on the other hand, is a much smaller program. It could conceivably fit onto a ROM with another product. We've forwarded your request to PCSG — but remember, T-View 80 is produced by one of PCSG's competitors in the Model 100 software market. Don't hold your breath. (For an analysis of the feasibility of fitting multiple programs into one ROM, see the Ultimate ROM review in this issue.)

It's completely possible to write a custom operating system for the Model 100 and plug it in on a ROM. All of the ROM-based applications on the market may be viewed as modifications of the Model 100's operating system — it's already happening. But Microsoft threw thousands of programmer-hours at the parts of the 100's operating system that control command interpretation, control of the LCD and communications. Sure, you could create a ROM that would duplicate these functions — but why?

Your suggestion that multiple insertions and removals could damage a ROM or a Model 100 hits home. We're starting to receive reports from users whose ROM sockets have been unaccountably damaged. No one's sure why yet, but until the dust settles you'd be wise to exercise special care when installing or removing accessory ROMs. —Ed.

PECULIAR PRINTER

Here's a correction for the July Portable 100/200. Page 37 inaccurately lists the Tandy TP-10 as battery-powered and compatible with the Model 100/Tandy 200.

The TP-10 is a thermal printer with a serial interface to be compatible with the Color Computer. It's AC-powered only and can't be used with the 100/200.

Robert Kuhn
CIS ID 71605,157

Tandy has verified your correction. Thanks for setting us straight. —Ed.

ERROR MESSAGE BLUES

Never have I read a magazine more closely, word for word, than Portable 100/200 — and I mean ads as well as articles. It's a great resource. I'm hoping you can help me.

I bought a copy of *The Model 100 Book* (Osborne/McGraw-Hill). On page 160 there is an alarm clock program. I've typed the program into my 100 and when I try running it, keep getting a UL error in line 30. I've tried retyping the line many times, as well as going back to the beginning and typing the first six or eight lines, but when I try to run it I always get *?UL 30*. The manual says this is an undefined line. But that doesn't tell me how to correct it and make the program work. I'm sure I'm doing something wrong — but what?

I have many books about the Model 100, many with programs, many giving guidance on everything relating to the 100. But what I need is a book or guide that goes deeply into the whole question of errors. What they are, what causes them and most of all — how to correct them. This last is where all the books fall down. They give you a few hints on debugging and error trapping, but nothing like real instruction on the entire error system or what to do when errors appear in your program.

What is an undefined line? How do I define it? What does defining a line mean?

Peter Olwyler
New York, New York

We understand your frustration. Your first step is: Take a deep breath and relax. You're not alone.

Programming is an art, not a science. Authors in the programming field try in vain to remember the moment before a certain concept "clicked." Can you remember what words looked like before you could read?

We don't have a copy of your book handy, but you've given us enough information to make a diagnosis. Line 30 of the alarm clock program passes control to another line of the program — probably with a GOTO or a GOSUB command. It may say 30 IF X = 0 THEN (GOTO) 100 *(the GOTO is optional in this case).*

The ?UL 30 error means your program doesn't have a line 100. Your program is trying to pass control to a line that doesn't exist (i.e., hasn't yet been defined).

Some programs pass control to any of several lines depending on certain criteria. These programs contain code like 30 ON F GOSUB 10, 20, 30, 40 *or* 30 IF X = 0 THEN 100 ELSE 200. *In these cases, you'll get a ?UL error only if the criteria are met. That is, sometimes your program will work perfectly and sometimes it will halt execution and beep at you.*

To solve a ?UL error, examine the line that gave you the error. You've either mistyped the line and sent program control to a non-existent line number, or you've mistyped the line number of the section of code that should execute next.

If the line that gives you the error is 30 IF X = 0 THEN 100, *you can always eliminate the ?UL error by typing a line 100, even if it's a non-functional piece of code like* 100 REM. *But this probably won't make the program do what you want it to do. Your best result is to type in the program again, taking special care to make sure the line numbers are correct. —Ed.*

200 A STEPCHILD?

I have been an avid aficionado of your magazine since its inception. About three or four months ago I traded up to a Tandy 200, donating my Model 100 to Good Will Inc. of Florida. It is my understanding from the officials at Good Will that my Model 100 is now being used to help youth and adults with speech difficulties.

My Tandy 200 has exceeded expectations. It is truly a work of art. However, I am beginning to regret my donation in spite of its altruistic overtones.

In the last issue you devote many pages to the adaptation of the Acroatix calendar motif to the Model 100 — but not a blessed word or even a nod of recognition to my 200.

Why? Is the Tandy 200 a stepchild?

Maurice H. Silk, M.D.
Kingston, New York

According to Acroatix officials, TMPC is not yet available for the Tandy 200. —Ed.

DIAL 1 (800) 732-5012 TO ORDER

MORE MEMORY

FOR YOUR TANDY 200 OR MODEL 100

TANDY 200
24K Expansion Modules only $129

CALL FOR LATEST PRICE

Two for $249

Expand your computer's memory in less than 5 minutes and save over $100 on each expansion module.

Take a dime and pop open the hatch, plug-in one or two of our 24K expansion modules, and increase your memory up to 72K.

Do you care about quality?

Whether you are writing a letter at home or projecting the corporate sales—the reliability of your memory is paramount. Purple Computing's 24K modules are designed specifically for the TANDY 200 and use the highest quality components.

- Uses the lowest power and most reliable memory chips available.
- Pins are made specifically for the sockets used in the TANDY 200.

MODEL 100, NEC PC-8201A AND OLIVETTI M10
8K Expansion Modules $35

Three for $99

Expand your computer's memory with our most popular product. Tens of thousands sold. Now you can save even more than before. You get the same high quality part and great service now at reduced prices.

- Easy to follow installation instructions included
- 30-day absolute satisfaction money back guarantee.
- Two-year replacement warranty.

Tandy 200 24K Expansion Modules $129 (#T200R)

Model 100 8K Expansion Modules $35 (#M1008K)

PC-8201A 8K Expansion Modules $35 (#NEC8K)

VARs, OEMs, Corporate users call for quantity prices.

It's Easy To Order By Phone—Just Call

1 (800) 732-5012 TOLL FREE.

Orders only (8am.-5pm. PST)

(805) 987-4788

For orders in California and for Customer Service

FREE SHIPPING, HANDLING, AND INSURANCE

for Cont. USA. Ordered today—shipped tomorrow UPS. 30-day total satisfaction, money back guarantee. We honor Visa, M/C, AmExp, and Money Orders. Checks held 3 weeks. California residents add 6% tax.

In a hurry? Ask for UPS BLUE (2-day air) only $4/order or UPS RED (1-day air) only $12/order.

**TANDY 200 is a trademark of TANDY CORP.*

PURPLE COMPUTING

420 Constitution Ave., Camarillo, CA 93010

Circle 49 on Reader Service Card

ONE QUESTION, ONE ANSWER

I am the proud owner of a Model 100 and have taken advantage of Portable Computer Support Group's 64K expansion module. In my mind that gives me more potential power than the Tandy 200, especially with PCSG's Lucid ROM installed.

But that's not why I'm writing. I need help or advice on what I may be doing wrong after entering and running CALNDR by R.E. Mendenhall (Portable 100/200, July 1985).

The program is very handy. It takes little memory and it's easy to use. Many programs are more detailed than they need to be; for those of us who just want to be organized it fits the bill quite well.

I've loaded the program exactly as printed, but when I try to call up a list of appointments I get a Bad File Number (?BN) error in line 780. This has caused me considerable frustration — I can't figure out what I'm doing wrong. I thought I had mistyped it so I double-checked it and even compared it with the format listed in the manual as well as other programs that call for the same operation.

Can you give me any suggestions?

In return, I've made a small addition to the program for your other readers. It's not much, but it adds a little. Change line 590 to:

590 IF INKEY$ = "" THEN GOSUB 1030: PRINT @265, CHR$(27) "p"; T$; CHR$(27) "q": GOTO 590

Then add the lines:

1030 PM$ = " am ": H = VAL (LEFT$(TIME$,2)): IF H >= 12 THEM PM$ = " pm ": IF H > 12 THEN H = H - 12

1040 T$ = STR$(H) + RIGHT$ (TIME$,6) + PM$: RETURN

This will display the time right under the key designations. I hope you can help me and that someone will find use for the display addition.

Gary L. Blodgett
Long Beach, California

Thanks for the program addition. You're right; the clock display is handy. But it sure eats up batteries!

Regarding your ?BN error: Check two factors. First, are you sure you opened NOTE.DO as file 1 (one), not file l (small "l")? That's a frequent source of ?BN errors.

Second, check to make sure that one of your other programs hasn't set MAXFILES to zero. Just to be on the safe side, you might want to set MAXFILES to 1 or more near line 100 in the initialization section of the program.

Thanks again. —Ed.

THE DVORAK OPTION

Do you know of any hardware or software support for the conversion of the Model 100 to the Dvorak keyboard system?

I own two Osbornes and understand that an OZROM hardware system permits redefinition of all keys. This could solve the problem for the Osborne. But what about Tandy portables?

Tedson J. Meyers
Washington, D.C.

It's theoretically possible to intercept keystrokes and translate them using a machine-language program like Michael Nugent's UNKEY routine in this issue. Then you could just pry up the keycaps on your 100 and replace them in the Dvorak arrangement.

Such an approach is cumbersome, however, and the program would be a long one. On a large scale, a ROM-based implementation would be practical. But the costs are probably prohibitively high for a single user.

We've asked Micro Demon's David Sumner to answer your question in a short article on Dvorak possibilities. Look for it in a month or two. In the meantime, you might work on a machine-language program based on Nugent's article and contact Polar Engineering at P.O. Box 7188, Nikiski, AK 99635, (907) 776-5529. Polar can help you explore firmware options. —Ed.

MORE PhD ARTICLES, PLEASE

Like many freelance writers, I have a pal: my Model 100. It's my best friend, as long as I don't mess with anything but TEXT. I can even manage to hook up a Radio Shack DMP 200 on a good day and persuade Write Plus to print out articles.

Although 99.5 percent of your magazine is as far beyond my ken as the Model 100 manual, I do enjoy browsing through it. I even bought Lucid after reading your recommendation, although I regret to report that, following your suggestion, I phoned Portable Computer Support Group and learned they have no plans to issue a Lucid that will work with a DVI.

While I have your attention, I'd like to suggest that you carry more PhD articles for kindergartners like me. (PhD = Push here, dummy!) The series by Danny Goodman was a step in the right direction.

I'd like to see articles in clear, nontechnical English on using interfaces, on sending signals to the printer and on using foreign-language characters. Why, for example, won't Write Plus accept the accented letters needed for French words?

Joan Melloan
Westfield, New Jersey

Thanks for the input. We're looking into establishing a beginners' section in the magazine for programming novices. This month's article by Dale Flanagan would probably fall into that category.

As for your Write Plus difficulty, it has to do with the way the 100 stores graphics and foreign-language characters in memory. These characters lie outside the usual 128 characters provided by the American Standard Code for Information Interchange, or ASCII. It's all explained in David Straayer's BLIST article in this issue.

We passed your letter around the office because we like the PhD concept. Then our associate publisher noticed your name and got us all wondering: Are you related to former editor John Mello? —Ed.

USER GROUP NOTES

Portable Computer Message System (PCMS) has moved. Sysop Jason Rich tells us the new address is c/o Software Riches, P.O. Box 1898, Babson Park, MA 02157-0910. Phone: (617) 239-5224.

PICOSHACK, the first computer bulletin board in Minnesota dedicated to laptops is now online. For more information write: Jim Radford, Sysop, 562 Tomlyn Avenue, Shoreview, MN 55126.

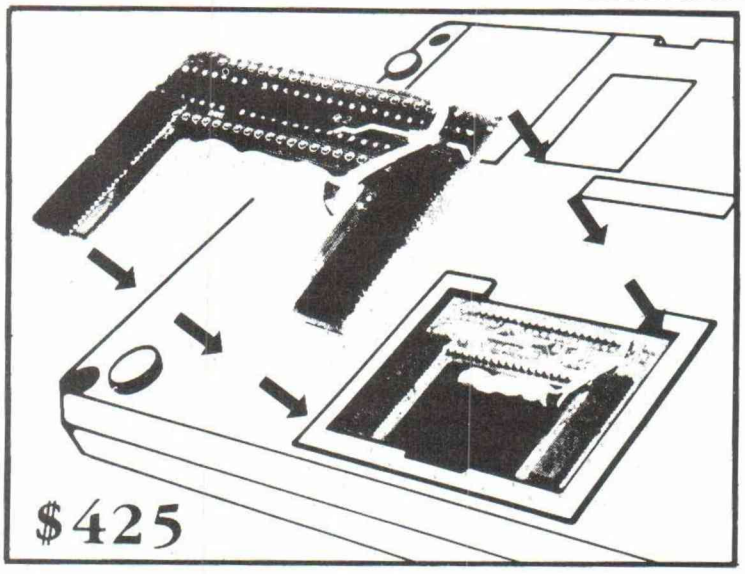
TRAVELING SOFTWARE NOT AT FAULT

A letter writer in the August GIGO cited Model 100 warranty problems caused by an 8K RAM upgrade chip purchased from Traveling Software of Seattle. Traveling Software's Neil Smith reminds us that the company has never offered a RAM upgrade, and suggests that the writer's problems were caused by another firms' products. We apologize for the error. — Ed. □

Give your Model 100

128K RAM

Installs as easily as plugging in a socket



Software included, transfers from bank to bank. Works like main menu! Includes powerful RAM Basic that lets programs store and access data from any other bank.

PCSG says: Satisfaction guaranteed or your money back within 30 days

As amazing as it seems you can upgrade your Model 100 to 128K of RAM in just 60 seconds.

It comes to you right out of the box looking just like the picture. You just open the little compartment on the back of your Model 100 with a quarter and it just pushes right into place. You can then put the cover back in its place.

You then have 4 banks of RAM of 32K each. The additional three banks also work just like your Main Menu.

You push a function key and you are in the second bank. Push again and you are in third, again, then fourth. Press it once again for your original bank.

It has its own built-in NiCad battery that recharges right from the Model 100 and its guaranteed for a full year.

What is really great is that you can copy a file from one bank to another with just a function key.

Each bank is like having another Model 100, and all the built-in programs as well as any snap-in ROM programs appear in all four banks and work the same way. Your widebar cursor moves from file to file and you access any file or run any program just by pressing ENTER.

What lets you copy any file from one bank to another is a snap-in ROM from PCSG called RAM+, that comes at no extra charge. It just pushes right into the little socket in that same compartment with the 96K expansion unit.

Not only does this firmware let you copy a file from bank to bank, but you can make a copy of any file within the same bank instantly with a function key. Great for Lucid spreadsheets!

Copy a file from bank to bank with a function key

You can also rename a file, or kill any file with just a function key. Plus you can do a whole lot of other useful things like setting the date, day and time with function key ease. You even have a function key that lets you use non-Radio Shack printers without having to make those tricky dipswitch settings.

RAM+ lets you cold start any one of your banks without affecting the other three. That means that anytime you want you can clean out a bank's entire memory, but leave intact all the files in the other banks.

What is also fantastic is that you don't have to have the ROM in place to use the additional RAM. Whenever you take out the snap-in ROM it leaves behind a tiny machine code program that lets you switch from bank to bank just by pressing ENTER.

This lets you use your ROM socket to snap-in other ROMS like LUCID spreadsheet, WRITE ROM text processor, or DISK+ ROM file transfer program, and use them in any or all four banks. All of these, by the way, are available from PCSG.

When you are ready to copy a file from one bank to another or use any of the other fantastic functions we talked about you can just snap the RAM+ ROM back into place.

Everybody that has this 128K system in their Model 100 is so excited, because it gives them four times the capacity and all banks work just like the Main Menu.

And what has made a lot of people happy is that the system bus, located in the same compartment, is left free for you to plug in a DVI or the Holmes Engineering/PCSG portable disk drive.

The ability to copy a file from bank to bank instantly with a function key, plus all of the other features make this RAM extension truly an engineering masterpiece.

Some people hesitate when they think of installing something, and then others are skeptical that any additional hardware could be as good as the Model 100 itself. That's why we sell these 96K expansions on a 30 day trial. Simply return it within 30 days for a full refund if you are not satisfied. Priced at $425. MC VISA COD.

1-214-351-0564

PORTABLE COMPUTER SUPPORT GROUP

11035 Harry Hines Blvd. No. 207, Dallas, Tx. 75229

Circle 33 on Reader Service Card

UPGRADING A STANDARD

An upgrade of the popular Epson FX-80 and FX-100 printers adds features — at little additional cost.

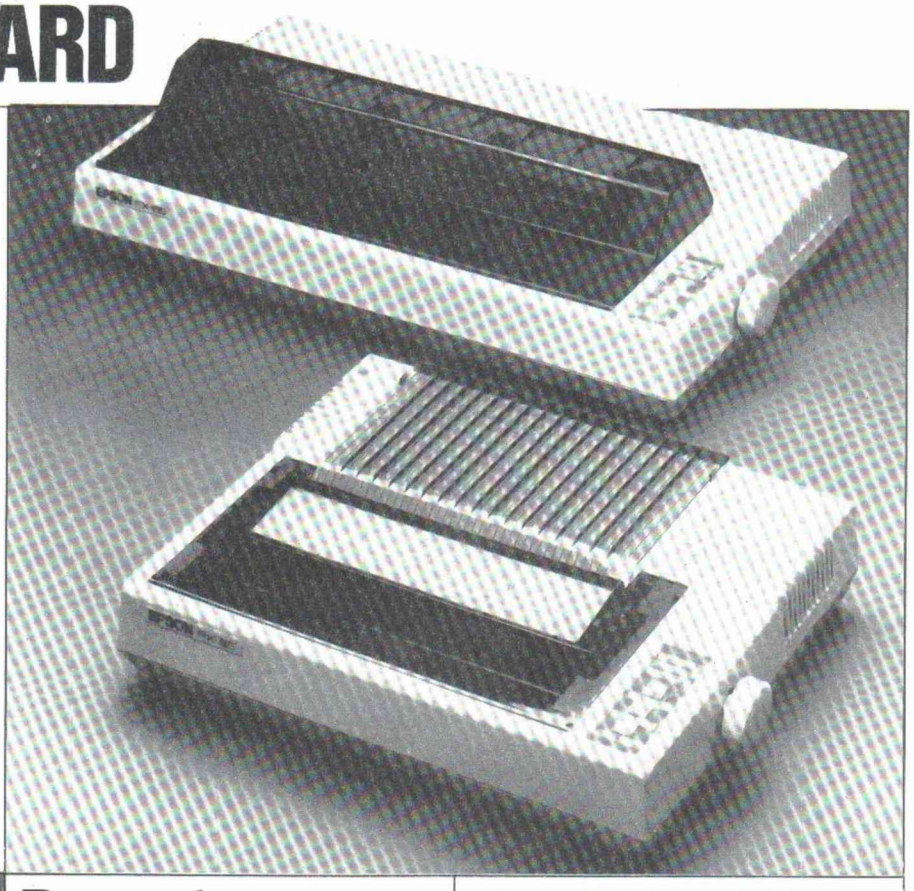
The new Epson FX-85 printer ($499) is an 80-column, 160-cps (characters per second) printer with a built-in 8K print buffer and IBM-compatible graphics. The Epson FX-185 ($699) is a 136-column version of the FX-85.

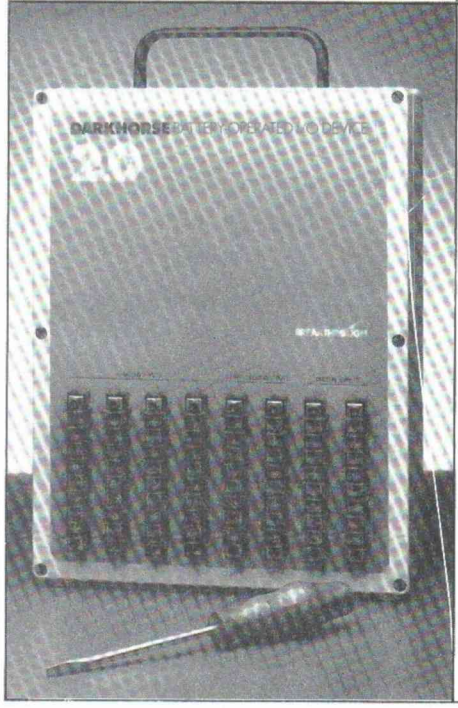
The major improvement: a firmware near-letter quality (NLQ) font, which prints at 32 cps. Both printers have both friction and pin-feed paper mechanisms. The FX-185 includes a paper tractor for smoother operation.

Cut-sheet feeders are also available. The feeder for the FX-85 has a suggested list price of $269. The accessory for the FX-185 lists for $319.

Owners can expand their FX-80, FX-100, FX-80+ and FX-100+ printers at local Epson dealers. The $85 upgrade kit gives the older printers all the features of the FX-85 and FX-185.

Contact Epson America's Computer Products Division, 2780 Lomita Blvd., Torrance, CA 90505, (800) 421-5426, (213) 539-9140 in California.
Circle No. 103





Rugged Instrumentation

Breakthrough, of Logan, Utah, has announced an environmentally-sealed analog-digital converter (ADC). The DarkHorse Model 2.0 ($697) is designed to withstand the hazards of dust, oil and moisture in the field, laboratory or factory.

The DarkHorse provides 15 analog input channels, each with a resolution of 100 microvolts in a four-volt range. The unit also has six digital outputs and four digital inputs, with a responsive voltage of zero or five volts DC.

A built-in temperature sensor responds in a -55 to 100 degree Celsius range, with one-degree accuracy.

Power for the DarkHorse is supplied by eight internal D cells or an external 12-volt power supply. The unit connects to the host computer via RS-232, at speeds between 300 and 9,600 bits per second (bps).

Contact Breakthrough, P.O. Box 230, Logan, UT 84321, (800) 235-6646 extension 667, (800) 235-6647 in California.
Circle No. 104

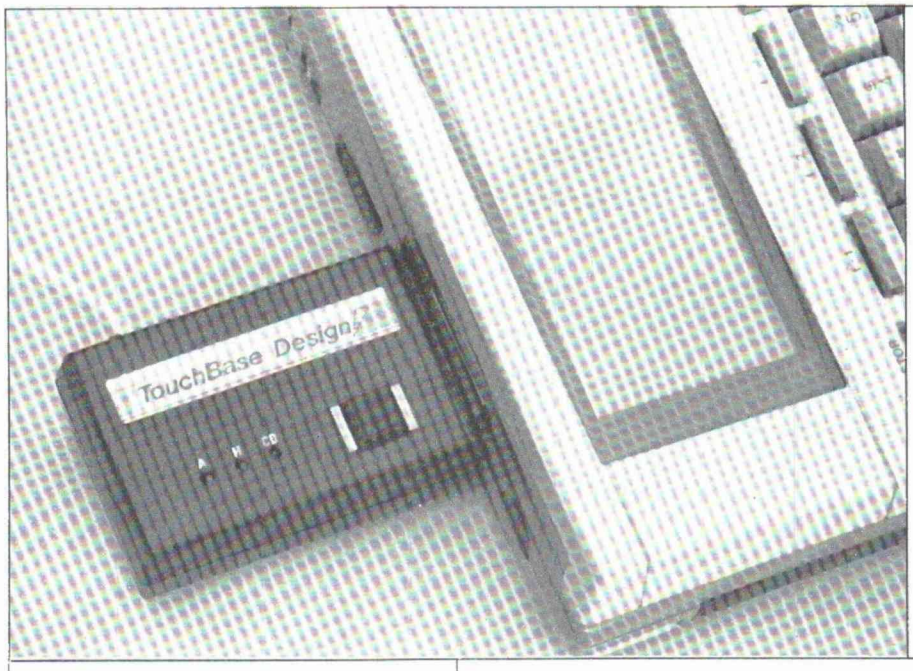
EPROM-It Yourself

Wavetek-Digelec's Model EP-824 EPROM and EEPROM programmer ($1,999) works with all currently available eraseable programmable read-only memories, as well as Intel single-chip microprocessors.

The programmer provides a built-in keyboard, as well as by remote control via RS-232.

The EP-824's EPROM erasing drawer holds nine chips. The Softpac EPROM cartridge holds four PROMs at 16K each.

Contact Wavetek-Digelec, 586 Weddell Drive No. 1, Sunnyvale, CA 94089, (408) 745-0722.
Circle No. 106



Travel Modem

TouchBase Designs of Los Angeles realizes travelers need modems as much as office-based users. The new TravelComm 1200 ($249) is a portable 300/1,200 bits-per-second (bps) battery-powered modem.

The modem features tone dialing and automatic transmission rate selection. It also shuts itself off at the end of session to save battery power.

The modem plugs directly into the computer's RS-232 port, and the telephone line cord attaches to the TravelComm 1200 — no special cables are required.

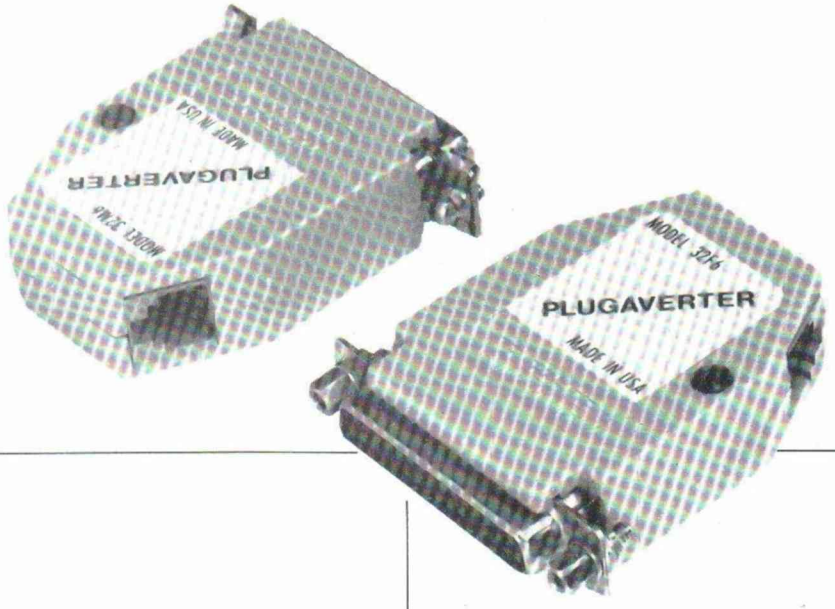
Contact TouchBase Design, 1447 So. Crest Drive, Los Angles, CA 90035, (213) 277-1208.
Circle No. 101

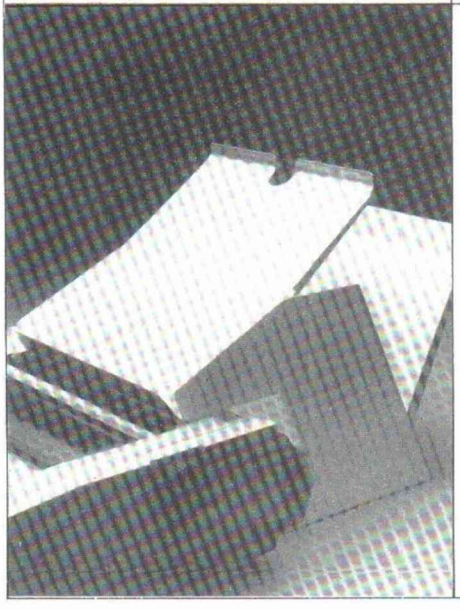
Paper Porter

The Porter System from Peri-Comp addresses the common problem: How should the printer paper feed?

The Porter (64.50 for 80-column printers, $74.50 for 136-column) holds five inches of fanfolded paper. A top rack takes 100 sheets of output without overflowing.

Contact Peri-Comp, P.O. 188, Lake Geneva, WI 53147, (414) 248-8585.
Circle No. 105





Simplify Cabling Operations

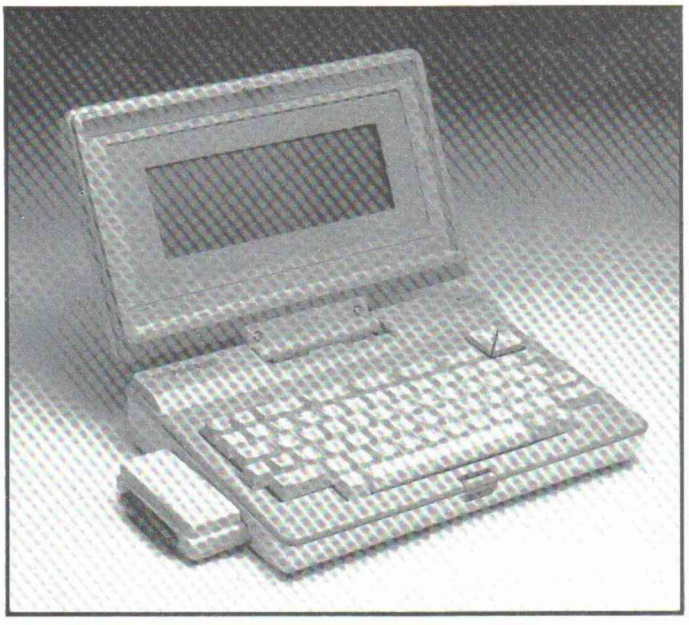
The Remark Datacom Division of Telebyte Technology offers products to simplify serial cabling situations.

The Model 32 Plugaverter ($13) allows less expensive wire to be used for RS-232 applications. The unit attaches to the DB-25 jack, which is the RS-232 connector on the Model 100 and most other microcomputers, and contains a jack for a modular plug.

Several variations on the Plugaverter theme are available, with the modular jack located at different positions on the connector. The standard unit uses RS-232 DB-25 pins 2, 3, 6, 7, 8 and 20; other configurations are also supported.

Modular cables for use with the Plugaverter and other modular applications are sold by the vendor. The Series 33 cables ($2.50 to $5) come in seven to 25-foot lengths, and have optional modular or spade-lug wires on one end.

Contact Teletype Technology, 270 E. Pulaski Road, Greenlawn, NY 11740, (516) 423-3232.
Circle No. 102



SIDESTAR

FOR THE NEC STARLET

The SideStar is a wallet-sized 128K RAM disk that gives the NEC Starlet more than just more memory, it adds new abilities to a great machine.

RUN THE POWERFUL SOFTWARE THAT REQUIRES MORE MEMORY

DBASE II and many of the powerful CP/M software packages require 64K mode and a disk drive which greatly reduces speed and portability. Supported by the Starlet's built-in ROM based CP/M operating system, the SideStar operates just like a disk and provides up to 100 times faster file access. Formated like a disk, a directory is created for 124 files and 124K of program and data storage. Free space is dynamically allocated in 1K blocks, so any file can be as large as needed, up to 124K. You need not give up portability to run large disk-based software.

EXPANSION CONNECTOR ADDS FLEXIBILITY

NEC Home Electronics participated in the development of the SideStar, including the specification of a bus connector which allows an additional cartridge (such as the NEC Disk/Video interface) to be plugged in behind the unit. This multi-cartridge configuration means that files can be loaded and saved efficiently and directly between the small, portable SideStar and the large, bulkier micro floppy disk unit.

FORGET ABOUT BATTERIES

A 1.2 Amp Lithium battery sealed inside powers the Sidestar up to seven years, into the next decade! You will not have to worry about batteries.

IS IT DISK—OR—IS IT RAM?

Think of the SideStar as a diskette. It's a self-contained, mass-storage device that requires no outside energy source. It can be used for transporting information between machines, long term storage and temporary storage. Unlike a diskette, it doesn't require a disk drive, power supply or cables, and it's very, very FAST! To the Starlet it is a disk.

A NO RISK OFFER

You can buy the SideStar for a 30-day evaluation. If you are not completely satisfied, return it within 30 days for a full and prompt refund. You get a two year warranty on parts and labor. If your SideStar should ever need service, we will fix or replace it within 72 hours.

INTRODUCTORY PRICE: $399

IT'S EASY TO ORDER BY PHONE—JUST CALL

1 (800) 732-5012 TOLL FREE

Orders only (8am.-5pm. PST)

In Calif. **(805) 987-4788**

For orders or customer service

FREE SHIPPING, HANDLING AND INSURANCE for Cont. USA. Ordered today—shipped tomorrow UPS. VISA, M/C, AmEx are welcome. Allow three weeks for checks to clear. Institutions & Corporations call for purchase order approval. California residents add 6% sales tax.

Price subject to change without notice. *DBASE II is a trademark of Ashton Tate.

PURPLE COMPUTING

420 Constitution Ave., Camarillo, CA 93010

Circle 45 on Reader Service Card

It's the Holmes Engineering/PCSG "chipmunk"

PORTABLE Disk Drive for the Model 100/200

$599
Includes a library of six powerful programs

Uses the main menu concept. You see the disk directory instantly, arranged on your M-100 screen like your main menu. Just move the widebar cursor and transfer files with a function key. You can run a file directly from the diskette with the ENTER key. Uses 3½" microfloppy diskettes that have a rigid plastic casing and a metal core. They're tough and nonflexible. You can carry several in a shirt pocket without damage. There's 358K on a diskette. Ten of these in your briefcase and you've got 3½ megabytes.

Drive weighs only three lbs. and it works directly from the 110 outlet and recharges at the same time. It recharges in six hours with thousands of pages transferred between charges. It's compact, with dimensions of 2¼" x 5½" x 7.5"; and fits easily into your briefcase along with your Model 100 or 200.

Machine code programs, BASIC programs, *Lucid* files and documents all are saved and retrieved with no protocol–instantly, ready to run.

In a special association, Holmes Engineering and PCSG have worked together combining the hardware knowledge of Holmes and the software expertise of PCSG. The result is a product that can only be regarded as excellent.

You see the disk directory instantly; works just like the main menu

Here is what is really exciting. The portable disk drive has Random Access. Included as part of the operating system in the drive (ROM) is a very powerful disk BASIC.

This means that you can have BASIC programs that will access the diskette and read and write records directly on the diskette.

Just imagine yourself with this kind of capability.

Database – The portable disk drive stores your mailing list, inventory items, part #s and descriptions or any other data that you need to recall.

358K on a diskette

Invoice (purchase order) – At the touch of a button you can print out your sequentially numbered, professionally done invoices. This is truly professional invoicing capability.

Purchase orders are just as easy.

Sort – This excellent utility allows you rapid sorting of any records you have compiled. You can write the newly sorted list back in the same file on the diskette or to a new file.

Telecom interface – If you are a user that likes to access other computers or databases (for example CompuServe) by telephone then this powerful facility alone is worth the price of the disk drive. You can automatically download and upload information onto the diskette.

Calendar – Everyone who has seen this program has said, "This is the first calendar/diary/scheduler on any computer anywhere that I can use. It is so functional."

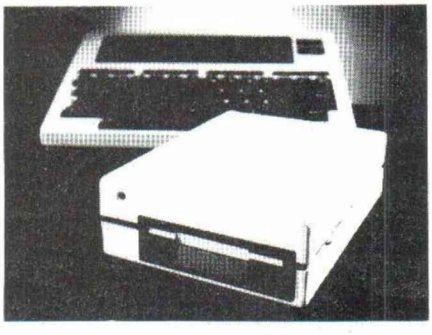
The calendar program is usable for two reasons, first it is designed correctly, and second you have the memory (358K) on the diskette to log and access a tremendous amount of notes over a long period of time.

Personal Finance Manager – This wonderful program truly lets you keep track of your finances.

All your records are kept on the diskette. Bank accounts (checking and savings) and charge accounts such as MasterCard and Visa.

We at PCSG believe we have the ultimate Model 100 system, the Portable Disk Drive plus the *Lucid* spreadsheet on snap-in ROM, *Write ROM* word processing and the new 64K RAM expansion now available from PCSG.

We want you to find out for yourself at no risk. If you aren't totally satisfied within 30 days, simply return the disk drive for a full refund. Priced at $599.95, including the software library. MasterCard, Visa, COD.



PORTABLE
COMPUTER
SUPPORT
GROUP

1-214-351-0564

11035 Harry Hines Blvd., #207
Dallas, Texas 75229

PCSG offers a 30-day/money-back trial!

Circle 37 on Reader Service Card

Getting Better All The Time

Program listers are a dime a dozen. But every once in a while someone comes up with one that's just a little bit better.

BY DAVID H. STRAAYER

What? Another BASIC program lister? Well, perhaps we can justify this one because of the interesting things we can learn about Model 100/Tandy 200 BASIC. Or maybe you'll find it has some features you want and haven't seen before.

There are good reasons for the huge number of program listers that have been contributed to the public domain or printed in *Portable 100/200*.

Tandy portables *do* have a built-in option, of course — Shift-Print. But this function has a couple of shortcomings. First, lines longer than 80 characters print off the edge of your paper. Second, programs longer than 66 lines print over the perforations of your paper — or off the end of the page if you're hand-feeding cut sheets into your printer.

These factors alone are inducement enough to make even the occasional programmer long for a better solution. But there are other desirable features, like putting the name of the program on the top of each page, and maybe the time and day too.

Finally, a BASIC program lister should make it easier to read your code by reflecting its structure.

Given a couple undisturbed hours, even novice programmers can write formatting programs that print highly readable listings based on BASIC programs stored as (or converted to) .DO files.

BLIST is different. Contributed by David H. Straayer of Colton, Oregon, it scans programs at runtime in their binary form — without converting them to .DO files. Just 1,308 bytes long, BLIST is much faster than other program formatters. And since it doesn't create a duplicate copy of the program as a .DO file, it requires far less random-access memory (RAM).

All in all, it's one of the slickest pieces of reader-contributed code we've seen all year. We're happy to share it with you. —*The Editors*

FIRST STOP: INITIALIZATION

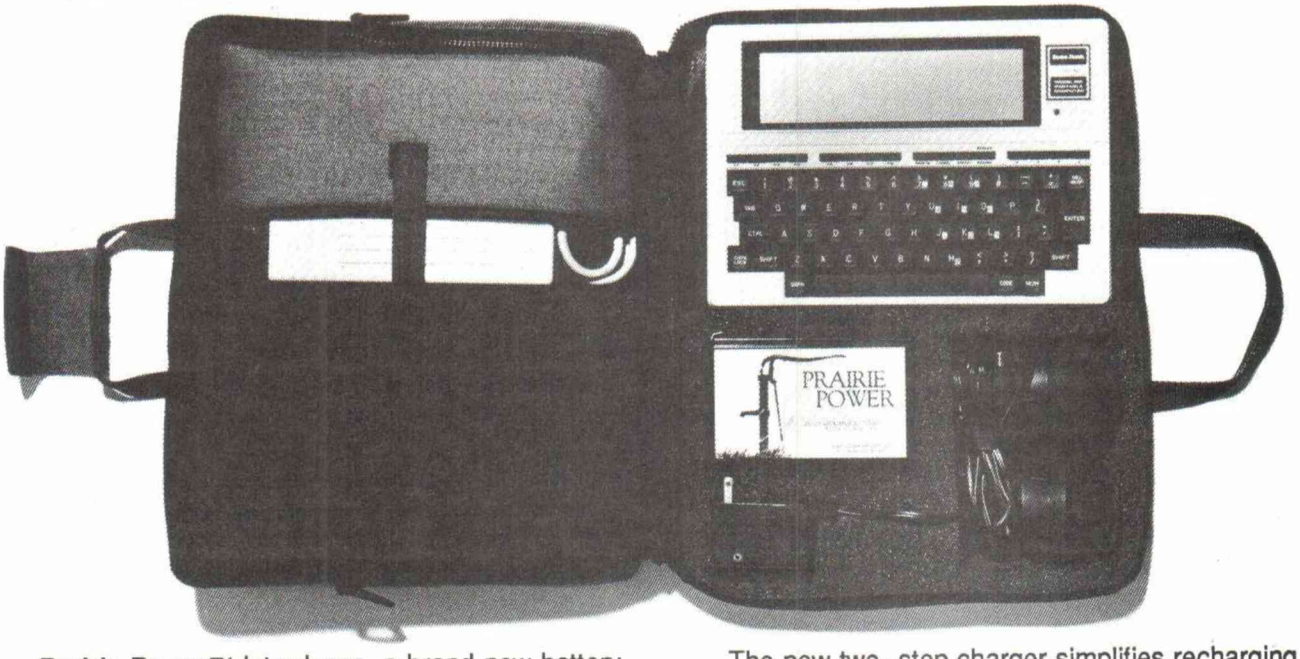
In line 1, we do some DIMensioning. Array KY% is a list of pointers to BASIC keywords in read-only memory (ROM). These will be needed in order to read the .BA image. T$ is an array of strings which should begin a new line in the listing in order to make the print-out easier to read. W is the maximum line width — it's the logical place to start fiddling around if you want to modify the program.

The rest of line 1 requires some explanation.

Perhaps you've noticed that the "year" part of the date on your Model 100 occasionally gets inexplicably changed. If so, it's probably due to a collision between an interrupt from the internal clock and an interrupt from the serial port.

We sneak around this bug by using a file called YEAR.DO that contains just four characters: 8, 5 and the two-character end-of-line string (carriage return plus linefeed). Next New Year's Eve I'll edit my copy of YEAR.DO and change it to 86. Meanwhile, any program that really cares about the date will read YEAR.DO to get the correct year. I've been using this little trick for quite some time now, and it works like a charm. If you forget to put in a YEAR.DO file, the OPEN statement will give you an error. Just use TEXT to create a file named YEAR with 85 (or 86 or 87. . .) in it and you're ready to go.

The Ultimate Battery System for your Portable Computer



Prairie Power™ introduces a brand new battery, charger and carrying case that greatly extends the power and efficiency for lap-top computers such as the **Tandy 100 & 200, NEC 8201, 8401, Apple IIc, Epson and Sord.**

The heart of the **Prairie Power** sytem is the lightweight rechargeable battery pack that can, for example, power the Tandy 200 up to 160 hours. By comparison, its "AA's" last a mere 10-15 hours. By the time a Tandy 100 has used up 10 packs of "AA's" **Prairie Power** is ready for its first charge (180 hrs.) Our new battery system can even power the Apple IIc for 8 hours and contains a sophisticated circuit to warn when the battery is low.

The new two--step charger simplifies recharging by showing when the battery is charged and then switches to standby. You always know the battery is charged and ready-to-go, whether for emergencies or long trips on the road. Besides the assuredness and efficiency of a powerful battery system, **Prairie Power** will pay for itself over the cost of "AA's".

Best of all, the battery system and computer are housed in a handsome water-resistant padded Cordura case protected by support panels and thick foam inserts that hold computer, battery, charger, cables and accessories. A padded pocket can also hold a flat-panel screen, small recorder or 3.5 "drive and manuals. **Call our toll-free number today!**

A Compact System!

The Prairie Power system fits under any airline seat and measures only (17x13x5"). With Tandy 100, battery, case and charger, weight is only 12 lbs. Apple IIc system weighs 20 lbs. Battery is fully rechargeable, sealed, maintenance free, useable in any position and air transportable. Never again worry over shelf-worn "AA's"

PRAIRIE POWER SYSTEMS, INC.™

Prairie Power Systems, Inc.
768 Twelve Oaks Center
15500 Wayzata Blvd,
Wayzata, MN 612-475-1945

Model 10 Battery ($59.95) Cable ($19.95)
Two-Step Charger ($55.95) Cordura Case ($99.95)
M /C, Visa, Check, MO $US only.

Total System Price $169.95*

Toll-free1-800-435-8721

**Apple 12-volt system ($249.95) Freight extra ($6.00)*

Of course, if you've exterminated the time-keeping bug with software of your own, you can always modify BLIST to use the last two characters of DATE$.

The arrays BA$ and AD are DIMensioned at 21; they will contain the names and addresses of all the programs in your 100's memory. Since 21 is the maximum, the arrays needn't be any larger.

In lines 2 through 5 the array KY% is loaded with the addresses of BASIC keywords in the 100's ROM. For more information, see the book *Hidden Powers of the TRS-80 Model 100* by Christopher L. Morgan (The Waite Group, 1984).

Lines 6 through 8 load the names of BASIC files into the array BA$(1. . .21) and their starting locations in RAM into AD(1. . .21). The program first checks to make sure the file type is right, then uses POKE to put the name into F$. We could PEEK the characters out one by one, but this is a little faster.

USER INTERACTION

Line 9 clears the screen, prints the names of all the programs in memory and prompts the user for the name of a file to print. Line 10 converts the user's response to capital letters so it can be compared to the internal names, which are always stored in caps.

Line 11 adds the correct number of spaces to make the file name six characters long. This makes comparisons easy because all the names in BA$ are exactly six characters long, with blanks at the end for short file names. Then line 11 searches through the files to find the user-specified name. If you wanted to be able to abbreviate the file name, you could add some intelligence here.

Line 12 processes the case of a mis-entered name. We beep the beeper and try again.

In line 13 things begin to get tricky. In most formatting programs, this is where we would read the .DO image of the BASIC program file. Since we're leaving the program in BASIC, we can't read it using conventional methods. Instead, we just PEEK around in RAM with pointers. IP is our statement pointer, and in line 13 we set it to the initial address for the file.

BASIC programs are stored in a *linked-list* structure. If a BASIC program isn't the current program running, all the statements follow each other in a nice neat line — like cows going home to the barn for milking.

But when you edit a BASIC program, individual statements may move around in memory. Therefore, the first two bytes of a BASIC statement are a pointer to the next statement. That's the *link*.

In line 14 we compute the address of the next statement (NX for next) and compare it to 0 — because the last statement in a BASIC program has a null pointer (0). When we get to a statement with a null pointer, we know we're done. We clear the screen and prompt for another file to print. This saves time, because the code between lines 2 and 8 takes about 10 seconds to run and we wouldn't want to repeat that wait with each file if we're listing more than one.

Line 15 computes the line number, which is stored internally in binary at the beginning of each line. If a statement number appears later in a statement — as a referenced line in a GOTO command, for instance — it appears in character form. This is why renumbering a BASIC program can make it somewhat shorter.

In line 15 we also begin to assemble L$, which will be a BASIC program line. The MID$ trick removes the initial blank STR$ produces on positive numbers, and we put a blank after the line number because the one space we're used to seeing in BASIC code isn't stored in the file.

In lines 29 to 36 we assemble the rest of the line. The variable JP is a pointer to keep track of where we are.

There are two kinds of data in a BASIC statement: keywords and other. The *other* data — strings, statement labels, variable names — are from the "usual" part of the American Standard Code for Information Interchange (ASCII) table. That is, they all have ASCII values of less than 128. This data is put in L$ in line 17.

DETOUR THROUGH ASCII LAND

Characters with ASCII values greater than 127 represent BASIC keywords, and we look them up in the table that's in ROM between locations 128 and 607 (decimal).

BASIC keywords are stored in ROM in a funny way. The first character of

```
1 CLEAR1000
    :DIMKY%(256)
    :Q$=CHR$(34)
    :DIMT$(3)
    :T$(1)=":"
    :T$(2)="THEN"
    :T$(3)="ELSE"
    :W=72
    :OPEN"YEAR.DO"FORINPUTAS1
    :INPUT#1,YR$
    :CLOSE1
    :DATE$=LEFT$(DATE$,6)+RIGHT$(YR$,2)
    :H$=SPACE$(10)+DATE$+SPACE$(5)+TIME$+SPACE$(5)
    :DIMBA$(21),AD(21)
    :N=0
2 I%=0
    :J%=127
    :KY%(128)=608
3 J%=J%+1
    :IFJ%>607
    THEN6
4 IFPEEK(J%)>127
    THENI%=I%+1
    :KY%(I%)=J%
5 GOTO3
6 FORI=63842TO64138STEP11
    :IFPEEK(I)<>128
    THEN8
7 N=N+1
    :AD(N)=PEEK(I+1)+256*PEEK(I+2)
    :F$="        "
    :FP=VARPTR(F$)
```

each keyword is OR'd with 128. This is how we know where one keyword leaves off and the next one begins.

ASCII allows definition of all alphabetic, numeric, mathematical and punctuation characters in seven bits — the values 0 to 127. The most significant bit, which has a value of 128, is ordinarily not used for character definition. Its most common use is as a parity bit in data transfer applications. In the Model 100, the eighth bit allows the user to display foreign-language characters and graphic symbols.

The eighth bit is also available to the system (and clever programmers) as a flag bit. That's precisely how the Model 100 uses it in the BASIC keyword table.

Take AND, for example. The ASCII value for A is 65 decimal. But it's stored in ROM as ASCII 193 (i.e., 65 plus 128). You can verify this by running this short program:

```
10 FOR I = 128 TO 607
20 PRINT CHR$(PEEK(I));
30 NEXT
```

As you can see, all the BASIC keywords are stored in these addresses. But their first characters appear on the screen as foreign-language characters or graphics. A second program yields a more familiar list:

```
10 FOR I = 128 TO 607
20 A = PEEK(I)
30 IF A > 127 THEN A = A-128
40 PRINT CHR$(A);
50 NEXT
```

This program masks out the offending eighth bit, returning characters from the "normal" ASCII range.

BACK ON TRACK

A similar algorithm is used to mask the high bit in the BASIC keyword table in program line 20. Note that lines 18 and 19 process two keywords in a special manner. These are the apostrophe (') comment delimiter and the keyword ELSE.

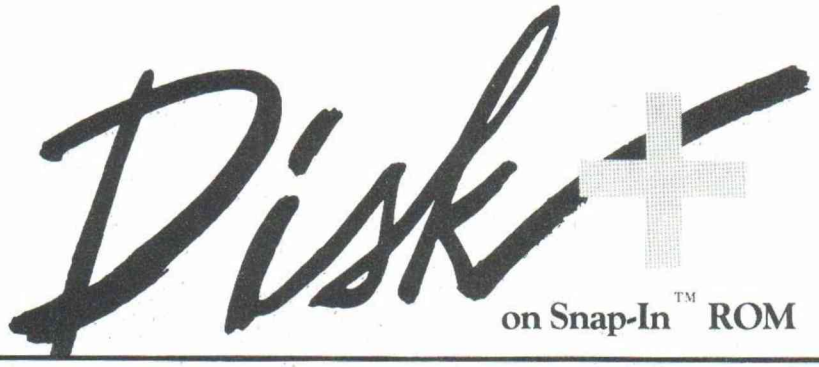
Line 21 copies in the remainder of the keyword if the keyword is longer than one character.

In line 22, we increment our pointer (JP) and go back to line 16. Notice that in line 16 we compare the next character to 0. Just as BASIC programs are delimited by a null pointer, BASIC lines are delimited by a null character (ASCII 0). This is how we know we've come to the end of a line and we're ready to print it.

In line 23 we have a line ready to print. We use two pointers — A and B —

```
    :IH%=INT((I+3)/256)
    :IL%=(I+3)-IH%*256
    :POKEFP+1,IL%
    :POKEFP+2,IH%
    :BA$(N)=F$
 8 NEXTI
 9 CLS
    :FORI=1TON
    :PRINTBA$(I);"  ";
    :NEXT
    :F$=""
    :PRINT
    :PRINT"Print which File";
    :INPUTF$
    :LOP=9999
    :IFF$=""
    THENMENU
10 FORI=1TOLEN(F$)
    :C=ASC(MID$(F$,I,1))
    :IFC>95ANDC<123
    THENMID$(F$,I,1)=CHR$(C-32)
11 NEXTI
    :F$=F$+SPACE$(6-LEN(F$))
    :FORI=1TON
    :IFF$=BA$(I)
    THEN13
12 NEXTI
    :BEEP
    :GOTO9
13 IP=AD(I)
14 NX=PEEK(IP+1)*256+PEEK(IP)
    :IFNX=0
    THEN9
15 ST=PEEK(IP+3)*256+PEEK(IP+2)
    :L$=MID$(STR$(ST),2)+" "
    :JP=IP+4
16 K%=PEEK(JP)
    :IFK%=0
    THEN23
17 IFK%<128
    THENL$=L$+CHR$(K%)
    :GOTO22
18 IFK%=145
    THENL$=LEFT$(L$,LEN(L$)-1)+"ELSE"
    :GOTO22
19 IFK%=255
    THENL$=LEFT$(L$,LEN(L$)-4)+"'"
    :GOTO22
20 J1%=KY%(K%-127)
    :J2%=KY%(K%-126)-1
    :L$=L$+CHR$(PEEK(J1%)-128)
    :IFJ1%+1>J2%
    THEN22
21 FORJ%=J1%+1TOJ2%
    :L$=L$+CHR$(PEEK(J%))
    :NEXT
```

USE YOUR DESKTOP COMPUTER AS A DISK DRIVE FOR YOUR M-100.



Try *Disk*+ for 30 days. If you aren't as excited as we are, return it for a full refund.

When we designed *Disk*+ we did it out of necessity. We wanted a way that we could just connect a Model 100 to our desktop computer with a cable and save files onto the desktop's disk drive. We wanted it to be so simple to use it would be self-explanatory.

Picture this. *Disk*+ comes to you on a Snap-in ROM and a diskette for your desktop. You take a quarter and open the little compartment on the back of your Model 100. Then you just press the ROM into the socket. *Disk*+ appears on your main menu just like a built-in.

You connect your Model 100 to your other computer using an RS232 cable (available from PCSG for $40).

You just place the *Disk*+ diskette into the desktop's drive and turn on the computer. It powers up automatically and says "awaiting command" on your desktop's screen. Then you just put the widebar cursor on the Model 100 main menu on *Disk*+ and press ENTER. You are shown your RAM files arranged just like the main menu.

To save a file to your other system's disk drive, you just move the widebar cursor to the file you want to save and press ENTER. It is saved instantly with no further action.

To look at the disk directory, you just press a function key on your Model 100. You see immediately the disk directory on your Model 100 screen, and it is arranged just like your Model 100's main menu.

To load a file from the diskette to your Model 100, you just move the widebar cursor to the file and press ENTER. The file is transferred to your Model 100's RAM instantly. You can press F8 and go back to the main menu, and the file you loaded from diskette is there, ready to use.

It is so nice to be able to keep your documents, programs (both BASIC and machine code) and *Lucid* spreadsheet files on the diskette, and bring them back when you need them. All files are ready to run or use with no changes or protocol by you.

If you have access to a desktop computer and don't have *Disk*+, then evidently we have done a poor job telling you about it.

All files and programs that you load or save, go over and come back exactly as they are supposed to be because of full error checking. This guaranteed integrity is really a comfort. *Disk*+ is wonderful in so many other ways. For example, you can do a "save all" of all your RAM files with just a touch of a function key. That group of files is saved on the diskette under a single filename with a .SD (for subdirectory) extension. Any time you want, you can bring back all those files at once, or just one or two if you like, again with one-button ease.

Disk+ takes up no RAM. That's zero bytes either for storing the program or for operating overhead.

What really excites most *Disk*+ users is text file cross compatibility. Your Model 100's text files are usable on your desktop computer, and your desktop's text files become Model 100 text files.

This means you can write something on your Model 100, and with *Disk*+ transfer it instantly to your desktop and start using it right away on your bigger computer. Or the way we like to work is to type in a document on the desktop computer and then transfer it to our Model 100 with *Disk*+. Then we print out the document, beautifully formatted, using WRITE ROM.

Disk+ works with just about every micro sold, from IBM PC and its clones, to all Radio Shack computers (yes, all), to Apple II, Kaypro, Epson and most CPM. Just ask us. More than likely, your computer is supported.

Incidentally, hundreds of Model 100 owners have gone to their Radio Shack stores and bought a color computer because it is so low priced, and with *Disk*+ they have an inexpensive disk drive.

And if that weren't enough, how about this: *Disk*+ also provides cross-compatibility between different computers like IBM, Apple or the Model 4 using the Model 100 as the intermediary device. Quite a feature!

The snap-in ROM is really great because you can use other ROMs like *Lucid* or WRITE ROM. They snap in and out as easily as an Atari game cartridge and you never lose your files in RAM.

Anyone who ever uses *Disk*+ simply can't do without it. But so many times we have had new users call us and say, "Wow! I had no idea when I ordered it that *Disk*+ would be so fantastic. I just couldn't believe that I could use my desktop computer's disk drive with my Model 100 just like it is another main menu."

That's why we sell *Disk*+ on a thirty-day trial. If you aren't completely satisfied, return it within thirty days for a full refund. Priced at $149.95 on Snap-in ROM. MasterCard, Visa or COD.

1-214-351-0564

PORTABLE COMPUTER SUPPORT GROUP

11035 Harry Hines Blvd., #207, Dallas, Texas 75229 • 1(214) 351-0564

PCSG provides hotline software support for the Model 100. Call us at 1-214-351-0564

Circle 36 on Reader Service Card

to step through the line, printing the single program line as several lines. These lines may be pieces of the statement that begin with a colon, a THEN or an ELSE, or pieces which have to be broken up to fit within our maximum column width.

Line 23 sets things up, and in line 24 we make sure there's something left to print. If not, we go back to line 14 for the next line.

In line 25 we search for each occurrence of a colon, THEN or ELSE. Suppose one of these is in a line as part of a string. Lines 26 and 27 perform an audit to make sure we don't make a false hit inside of quoted strings. The variable Q$ contains a question mark, and we use it to find strings within the line.

We don't bother to check to see if the colon, THEN or ELSE is part of a REM statement. This is a bug or a feature, depending on how you look at it. I sometimes REM out portions of code temporarily while I'm debugging, and this program lister then reflects the original structure of my code.

In line 30, we're almost ready to begin printing a line. First we check variable LOP to see how many lines we've already printed on the page. If we've already printed our limit or we're starting a new file, we force a page break by printing a form-feed character: ASCII 12. Most printers jump to the next page when they receive this character. If yours doesn't, you can change this code to print out enough blank lines to be at the start of the next page. Or you can prompt the operator to insert a new sheet of paper in hand-fed printers.

Line 30 also prints the header, H$. I like my listings to contain date, time and the name of the file I'm printing. If we just printed TIME$ and DATE$ here, different pages would have different times — that could get confusing if pages got separated. That's why we defined H$ earlier.

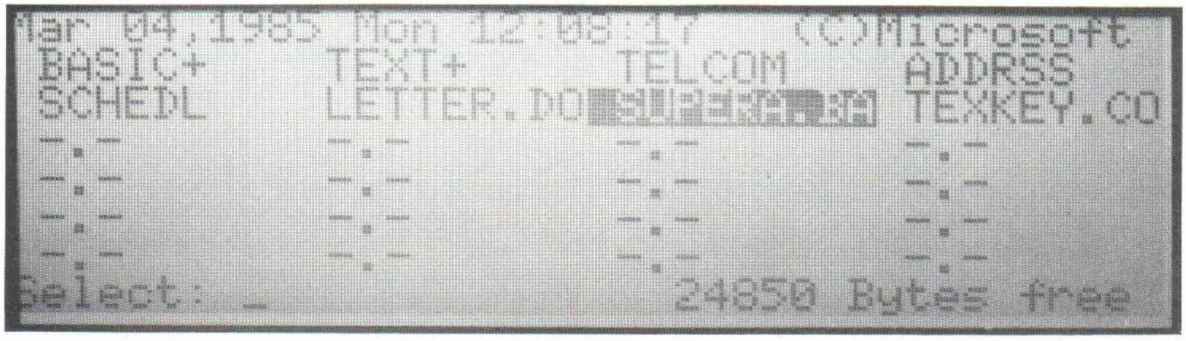
Line 31 prints out the line itself. The variable P keeps track of indentation. The first line of a statement isn't indented; subsequent lines are indented four spaces.

That's it. There are plenty of hooks here for adding and changing program features. You might want to change the headers, number the pages at the top or bottom, change the indentation level or change the maximum legal line width. Have fun. □

Please help us rate this article's value. If you've found it very valuable, circle 161 on the Reader Service Card. If it was moderately valuable, circle 162—and if it wasn't valuable to you, circle 163.

```
22 JP=JP+1
    :GOTO16
23 P=0
    :Z=W
    :B=0
24 A=B+1
    :K=A+1
    :IFA>LEN(L$)
    THENL$=""
    :IP=NX
    :GOTO14
25 Q=A+1
    :F=0
    :FORI=1TO3
    :G=INSTR(K,L$,T$(I))
    :IFF=0ORGANDG<F
    THENF=G
    :NEXT
    ELSENEXT
26 R=INSTR(Q,L$,Q$)
    :IFR=0ORR>F
    THEN28
27 S=INSTR(R+1,L$,Q$)
    :IFSANDS<F
    THENQ=S+1
    :GOTO26
    ELSEIFS
    THENK=S+1
    :GOTO25
28 C=A
    :IFF=0
    THENB=LEN(L$)
    ELSEB=F-1
29 IFC>B
    THEN24
    ELSEN=B-C+1
    :IFN>Z
    THENN=Z
30 IFLOP>50
    THENLPRINTCHR$(12)
    :LPRINTH$;F$+".BA"
    :LPRINT
    :LOP=0
31 LPRINTSPACE$(P)+MID$(L$,C,N)
    :P=4
    :Z=W-P
    :C=C+N
    :LOP=LOP+1
    :GOTO29
```

The Model 300 Is Here... You Already Own It!



SUPERA: Like Getting A Whole New Computer

In January 1983, Micro Demon introduced PRO AID. Suddenly Model 100 users found that their computer had capabilities far beyond their expectations. In his InfoWorld review of PRO AID, Reviewer Greg Springer, anticipating the announcement of the Model 200, wrote,

"PRO AID adds enough new capabilities to the current model that the wait for upgraded portability is made much easier and maybe even unnecessary."

Well, the Model 200 is here. But so is SUPERA! SUPERA takes a giant step past PRO AID, and adds a multitude of powerful new features to the Model 100, making it into the computer it was meant to be.

Once it is loaded, SUPERA works transparently to you and your programs. If it weren't for all the wonderful things it does, you would never even know it was there. If you want to remove it, SUPERA is as easy to take out as KILLing a BASIC program.

Unique Format

SUPERA loads and runs as if it were a BASIC program (it's really all machine language). Because of this unique format, SUPERA is compatible with most other software and hardware. It also only requires 4.3K of memory.

With SUPERA installed, every facet of your computer suddenly takes on an aura of new-found power.

In Text

While in TEXT you can

- Use a really fast and flexible search and replace function.
- Turn on a special type-over mode in which typed characters replace text instead of being inserted.
- Read a second file while editing a first.
- Use control keys to delete words and lines, to change the case of the character under the cursor, and to activate the paste key.
- Cause the display to scroll slowly up or down a line at a time.
- Redefine special keys to act as SHIFTed keys, control keys, or graphics characters.
- Turn the computer off without losing your place.
- Access special HELP files.
- Use 26 new macro function keys.

and more!

All of Supera's editing features become available whenever you enter EDIT mode to edit a BASIC program.

Super Function Keys

A great deal of SUPERA's power comes from its 26 macro function keys. These easy to define keys can be used in TEXT, TELCOM and BASIC. Each key generally defines a string of 14 characters, but in both BASIC and TEXT several keys can be concatenated into one so as to provide longer keys. The power of these function keys is hard to imagine until you try them.

In TEXT they can provide you with much more than just easy entry of boiler-plate strings. They can also be defined to automatically carry out complicated editing sequences.

In TELCOM these keys can save you money while accessing a data base such as Compuserve. They also save you the trouble of remembering such things as passwords, ID numbers, and special commands.

In BASIC use the keys to enter BASIC commands, and to facillitate typing in programs, to name just a few applications.

Moreover, SUPERA allows you to save and load entire sets of function keys by a single control key action. Thus you might keep a set of keys called TEXKEY.CO handy to use in TEXT, another keyset for BASIC, and yet another for TELCOM.

In BASIC

You get the 26 function keys, and also

- Single stroke access to the built-in software such as TEXT and TELCOM
- Automatic Line Numbers
- Control key activation of selected BASIC commands
- You can redefine special keys

In Menu

You can use control keys to kill and rename files.

Calculator Mode

From BASIC you can enter an entirely new mode called Calculator mode. While in this mode you can quickly and easily evaluate expressions. Calculator mode also provides several new BASIC functions.

Better Than PRO AID

It's true that PRO AID was a significant enhancement to the Model 100, but SUPERA goes far beyond it. In a message on the Compuserve Model 100 SIG, system operator Dave Thomas said,

"Now with SUPERA, the original PRO AID has been enhanced beyond belief... If there is a single, more useful utility on the market for the Model 100 than SUPERA, I haven't heard of it and t'would boggle my mind if t'was!!"

TEX PRO For Word Processing

TEX PRO consists of ALL the TEXT editing features of SUPERA, uses only 2.8K of memory, and sells for just $49.95. It may be the choice of those who seldom use their Model 100 except for word processing, and don't need all the additional power of SUPERA.

MICRO DEMON, INC.

For more information or to order, write to Micro Demon, Inc., P.O. Box 50162, Columbia, SC 29250. Or call 803-733-0980 anytime. SUPERA $79.95, TEX PRO $49.95. Add $3.00 for shipping. Visa and Mastercard are welcome.

?? DO YOU PROGRAM BASIC ??

The ONLY BASIC DEBUGGER combined with a powerful TEXT EDITOR on plug-in ROM

Now for the Tandy 200 and Nec 8201A !!

Cleuseau

BASIC Inspector and Text Editor

● Search and replace ● overwrite mode ● append the paste buffer ● capitalize or lowercase a section of text ● count bytes/words/lines ● verify a file ● Compress a BASIC program, eliminating all unnecessary text. ● Renumber, move, copy program lines. ● Stop your program anywhere, anytime to trace exactly where you've been, without losing variables. ● uses no RAM memory.

● MANY, MANY MORE FEATURES!!! ●

Cleuseau is designed to let the beginning programmer avoid the pitfalls of BASIC programming while providing the advanced programmer the ease of ***Cleuseau's*** many shortcuts.

$95 plus $4 First Class or $15 Express Mail

Cleuseau **+ ROM2**
in one ROM **$145**

VISA **CALL TO ORDER** MasterCard

Please specify for Model 100, Nec 8201A or Tandy 200.

The ROM Experts . . .
Our software and yours in ROM for the Model 100, Nec 8201A and Tandy 200.

Polar Engineering and Consulting
P.O. Box 7188, Nikiski, Alaska 99635 (907) 776-5529

One Cable Does All

Serial cables: You can't live with 'em and you can't live without 'em. But having to rely on just one would sure make life simpler.

By Jeffrey Green

For those of you frustrated by constantly switching serial cables to communicate between devices, take heart. Here's a simple do-it-yourself hardware tip that lets you use one cable for all your computing needs. Computer to modem or computer to computer — it's as easy as flicking a switch.

SOME BACKGROUND

The RS-232 interface is used in two types of serial devices: data communications equipment (DCEs) and data terminal equipment (DTEs). DCEs are computers and the like which *control* serial data transfer. DTEs such as modems and printers are under the command of DCEs.

Computers, and their DCE cohorts, send data via pin two of the common 25-pin serial connector. Information is received from remote devices over pin three.

Just the opposite is true with DTE equipment. Pin two receives information and pin three sends it.

Off the shelf, your standard RS-232 serial cable is constructed "straight through." Simply put, pin one on one end connects with pin one on the other, pin two with pin two and so on. Hence, the "talking" pin two of the DCE is tied to the listening pin of the DTE, and vice-versa. Communications takes place.

The problem comes when trying to connect two computers directly, without a modem. Since both DCEs are talking on pin two and listening on pin three, a straight cable just won't cut it. Both computers will be talking into thin air.

Enter *null-modem*. This special cable merely swaps pins two and three on one end — causing each snobbish computer to think the other is a lowly DTE. Both computers are satisfied and communications proceed accordingly.

Null-modem cables or adapters aren't much more expensive than straight cables. Depending on length, they go for about $25-30.

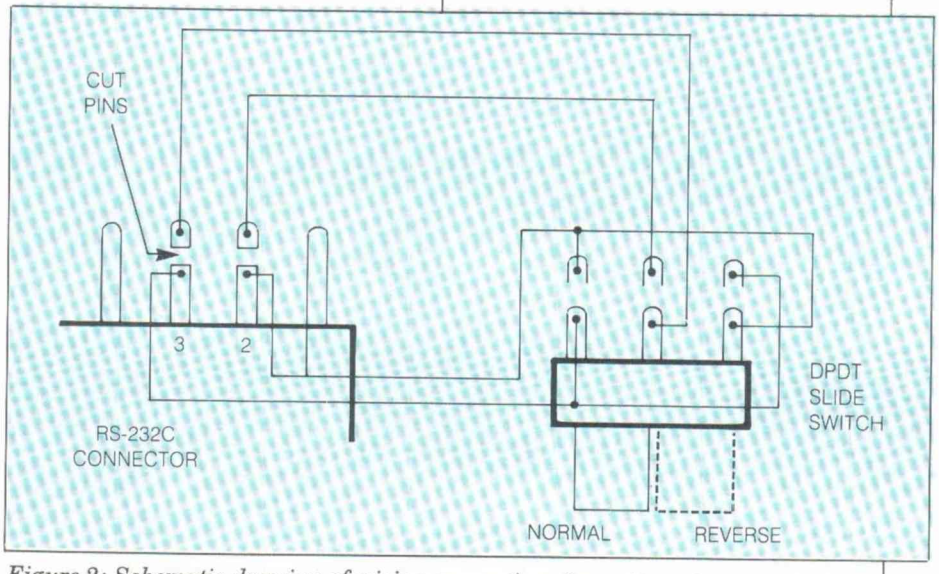


Figure 2: Schematic drawing of wiring connections for null-modem switch. Solder connections as shown before attaching switch to plastic plate.

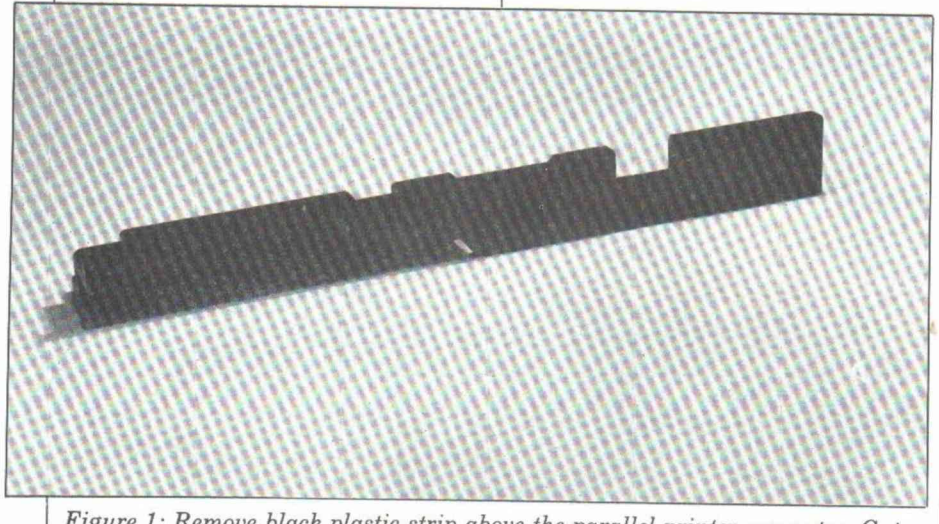


Figure 1: Remove black plastic strip above the parallel printer connector. Cut a slot in vertical face, wide enough for switch to slide freely.

Tandy laptops, however, are designed for portability. So why carry two cables when one will do?

FLICK MY SWITCH

An alternative to buying a null-modem cable is to install a manual switch that will make the pin swap *inside* your 100. In effect, you can change your laptop from a DCE to a DTE. All that's needed is a double-pole, double-throw (DPDT) slide switch, some wire and a few common tools.

WARNING: Opening your laptop's case and modifying the electronics will void its warrranty. If your skills with a soldering iron are weak at best, perhaps you should consult your local electronics repair shop.

KEEP IT CLEAN

Before starting, find a clean, soft work surface. Wooden tables on a hard floor are best — rugs and carpets attract dangerous static electricity.

Back up the computer's RAM onto cassette or disk. The modifications shouldn't affect memory, but why take chances? Also it's advisable, but not necessary, to remove the batteries.

Turn the computer over and remove the four corner screws. It should open like a book. Notice the two ribbon cables connecting the two halves. Be careful not to damage these.

Remove the black plastic strip (two screws) above the parallel printer connector — that's where we'll put the null-modem switch. Cover the exposed halves of the computer to keep dust and debris out.

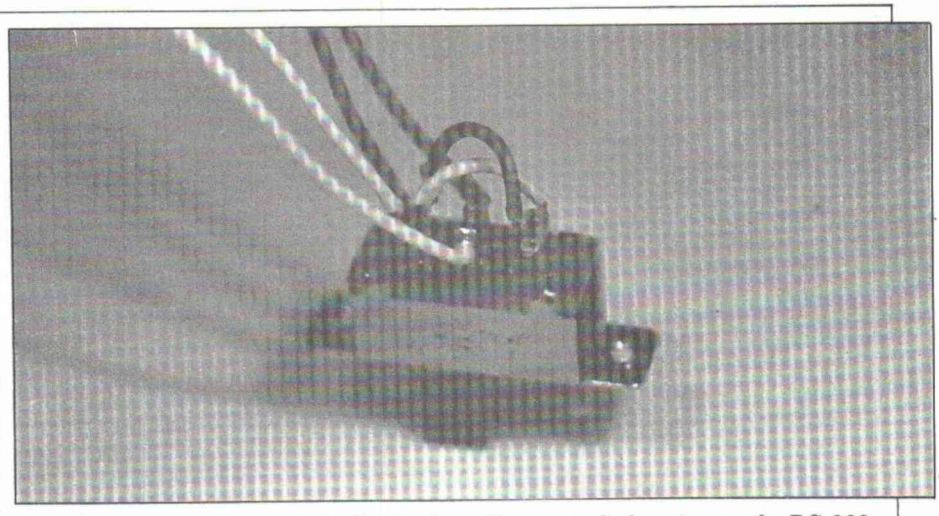


Figure 3: Null-modem switch all wired up. Next, attach the wires to the RS-232 connector.

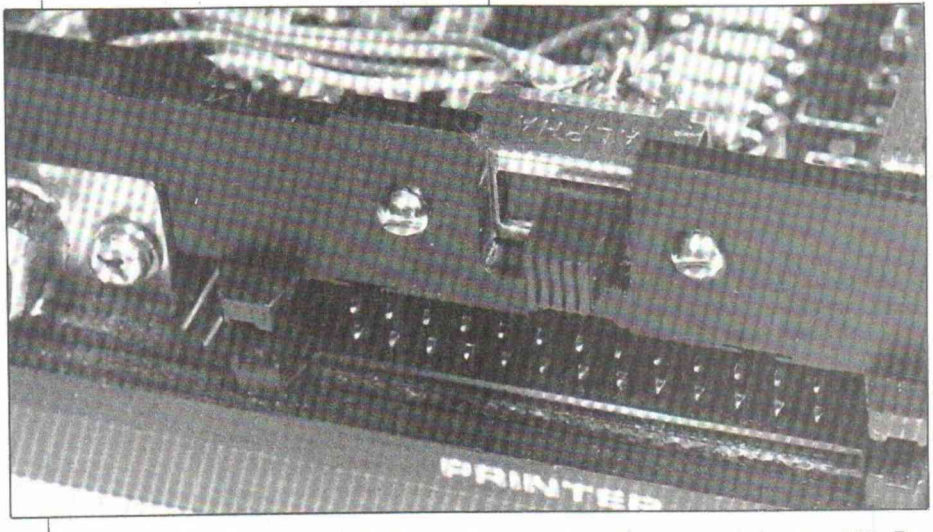


Figure 4: Screw the switch to the mounting plate and remount it in your 100. Be sure connections are secure. Clean up any debris before closing cover.

Cut a slot in the vertical face of the strip, midway between the mounting holes. The cut must be wide enough for the switch to slide freely. Mark and drill holes for the switch's mounting screws.

The top half of the case must be cut to leave room for the switch. Again, be careful no debris gets in the computer. While you're at it, you might cut slots above and below the RS-232 connector. Some cables have difficulty fitting into the space provided.

WIRE IT UP

Before attaching the switch to the plate, solder the connections as shown in Figure 2. You'll have to cut the internal connections to pins two and three first. This is tricky and must be done with care.

After soldering the internal wiring (use a small vise or locking pliers for ease) attach the wires to the RS-232 connector.

Finally, screw the switch to the mounting plate and remount it in your Model 100. Carefully close the case and insert the corner screws.

As a last step, label the new switch settings for normal DCE configuration and reversed, or DTE configuration.

Now you're ready to roll. One cable does all with just a flick of a switch. Whether you're using a 1,200 bits-per-second(bps) external modem, or want to download from a desktop at 9,600 bps, the cable in hand is all you need — every time. □

Please help us rate this article's value. If you've found it very valuable, circle 179 on the Reader Service Card. If it was moderately valuable, circle 180—and if it wasn't valuable to you, circle 181.

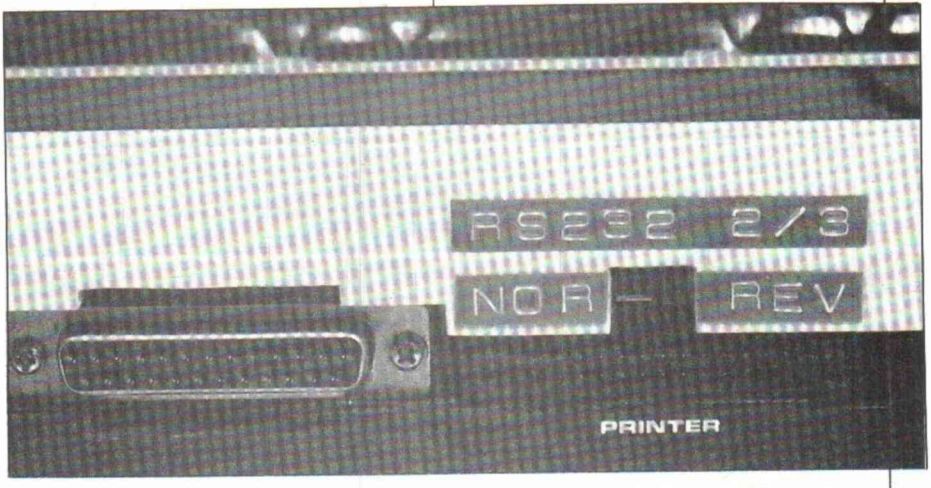


Figure 5: Last step, label the switch settings for normal (NOR), or DCE configuration and reversed (REV), or DTE configuration.

IS BIGGER BETTER?

Thinview
Intelligent display system
Axonix
417 Wakara Way
Salt Lake City, UT 84108
(800) 821-7093
$649
Circle No. 142

By CARL OPPEDAHL

How would you like a bigger screen for your Model 100? Maybe 80 characters by 25 lines? Yes? Then perhaps the Axonix Thinview is for you.

The Thinview comes in a sturdy beige and black plastic housing that nearly matches the Model 100. It is 11.75 inches wide (same as the Model 100), 6.5 inches tall and 1.5 inches thick. The Thinview and cable add about three pounds to the 3.75 pound Model 100. It runs on a built-in rechargeable nickel-cadmium (nicad) battery, and, when connected to the Model 100, can power itself up and down automatically with the computer.

Axonix says the design goal for the Thinview was to emulate, as closely as possible, video functions of the Radio Shack disk-video interface (DVI). As a result, the BASIC commands, such as SCREEN and WIDTH, which activate the DVI display if installed, are set up to activate Thinview. The command SCREEN 1 is supposed to make the Thinview into an expanded, but otherwise identical, replacement for the built-in LCD screen while the user is in BASIC, TEXT, ADDRSS, SCHEDL or TELCOM.

Thinview comes close, but is not identical to, the built-in LCD screen. For example, in BASIC, ADDRSS, SCHEDL and TELCOM, Thinview scrolls strangely. If you send lots of BASIC output to the Thinview screen, the output will scroll up the screen and disappear, just as on the built-in LCD screen. However, every so often a moment arrives when the screen clears completely and starts again at the top. This happens every few hundred lines. In many applications this really doesn't matter, but for a program which relies on the user being able to see previous lines of output, Thinview may be unusable. Imagine a program that prints several lines of information, follows it with a question, and asks you for your answer. The information may be lost in the screen clearing process. Similar problems crop up in ADDRSS and SCHEDL.

Where unbroken screen output is important, an ambitious BASIC programmer can avoid the problem by intentionally clearing the screen before the cursor reaches the lower right corner, and repainting anything that is required.

Changes your Model 100 into a totally different computer with capability you never thought possible.

Lucid™ on Snap-In™ ROM

Infoworld rated Lucid's performance "excellent"

PCSG says "Satisfaction Guaranteed or your money back within 30 days!"

LUCID© is here now. It is on a ROM cartridge that snaps into the compartment on the back of your Model 100. It takes no memory to load and no memory for operating overhead. That means you have the full 29.6k bytes free to store your data.

First, LUCID© is memory conserving. It will let you build a large spreadsheet – 255 row by 126 column capacity. You build huge spreadsheets in your Model 100's RAM that could consume 80 to 100K on a desktop computer.

Secondly, LUCID© is fast. LUCID© is so rapid, a 36 column corporate financial statement took less than 4 seconds to calculate.

Thirdly, LUCID© has features you won't find in most other spreadsheets. For example, when you type a label (text) it will cross column boundaries; in other words when you type a label or title it will appear as you type it irrespective of column or width. LUCID© also allows you to set column widths individually, and of course LUCID© has insert row and insert columns, as well as other standard features. LUCID© even lets your formulas refer to cells in other spreadsheet files.

Further, LUCID© has what no other spreadsheet has: Cut, Copy, and Paste. It uses the same keys as Cut and Paste in TEXT, but here's the difference: it takes all the formulas with it when you paste and they all automatically recalculate with the entire sheet.

And here is what is really amazing. You can copy or cut from one spreadsheet and paste into another spreadsheet or even a TEXT file.

LUCID©supports all BASIC math functions as well as Log, sine, cosine, tangent, exponentiation and other sophisticated math functions.

LUCID© has so many features that you will say "this is what I need in a spreadsheet", such as automatic prompting of an incorrectly typed-in formula showing just where the mistake was made.

LUCID© has expanded "go to" functions that remember and produce a windowing capability.

But perhaps most remarkable is that LUCID© is not only a spreadsheet but a program generator as well. First, LUCID© lets you protect all cells against entry or change, and then unprotect just the cells you want for someone else to use as input fields.

LUCID© will not only process values, but text input as well so that the facts other than numbers can be responded to. LUCID© has the ability for you to refer in a formula to cells containing words. This feature combines with the capacity of doing "if then" statements that work by doing table lookups against even massive X/Y charts of text or numerical information. You can produce a program that responds to inputs with no programming knowledge whatsoever.

You can prepare a report section in your spreadsheet with instructions to your user for printout, and they can produce a personalized printout that responds to their input. All your formulas and tables that did the calculations and provided the facts are invisible to that user. LUCID© is useful for doctors for patient questionnaires, troubleshooting technicians, purchase clerks, people doing job quotes, stores for customer workups, insurance agents and anybody who needs to process specific facts and numbers to produce a report based on those responses.

LUCID© comes with a manual that explains not only the characteristics of LUCID©, but will train you how to use a spreadsheet even if you have never seen one before. You are shown how to do budgets, forecasts, breakeven analysis amortizations and many other types of personal and business reports and calculations.

User friendly is such an over-used term in this industry, but a typical comment has been "I have never seen a spreadsheet that does so much, and yet LUCID© is so much easier and faster to use."

LUCID© is a result of a most exhaustive developmental effort in which PCSG's objective was to develop a spreadsheet that was better than the state-of-the-art. We are so pleased because LUCID© provides for the Model 100 spreadsheet capability you cannot equal on a desktop computer.

LUCID© is, in our opinion and that of those who have examined it, a breakthrough. We sell it on a 30 day trial. If you are not completely satisfied, return it within 30 days for a full refund. Priced at $149.95, on snap-in ROM. MasterCard, Visa or COD.

First with software for the Model 100

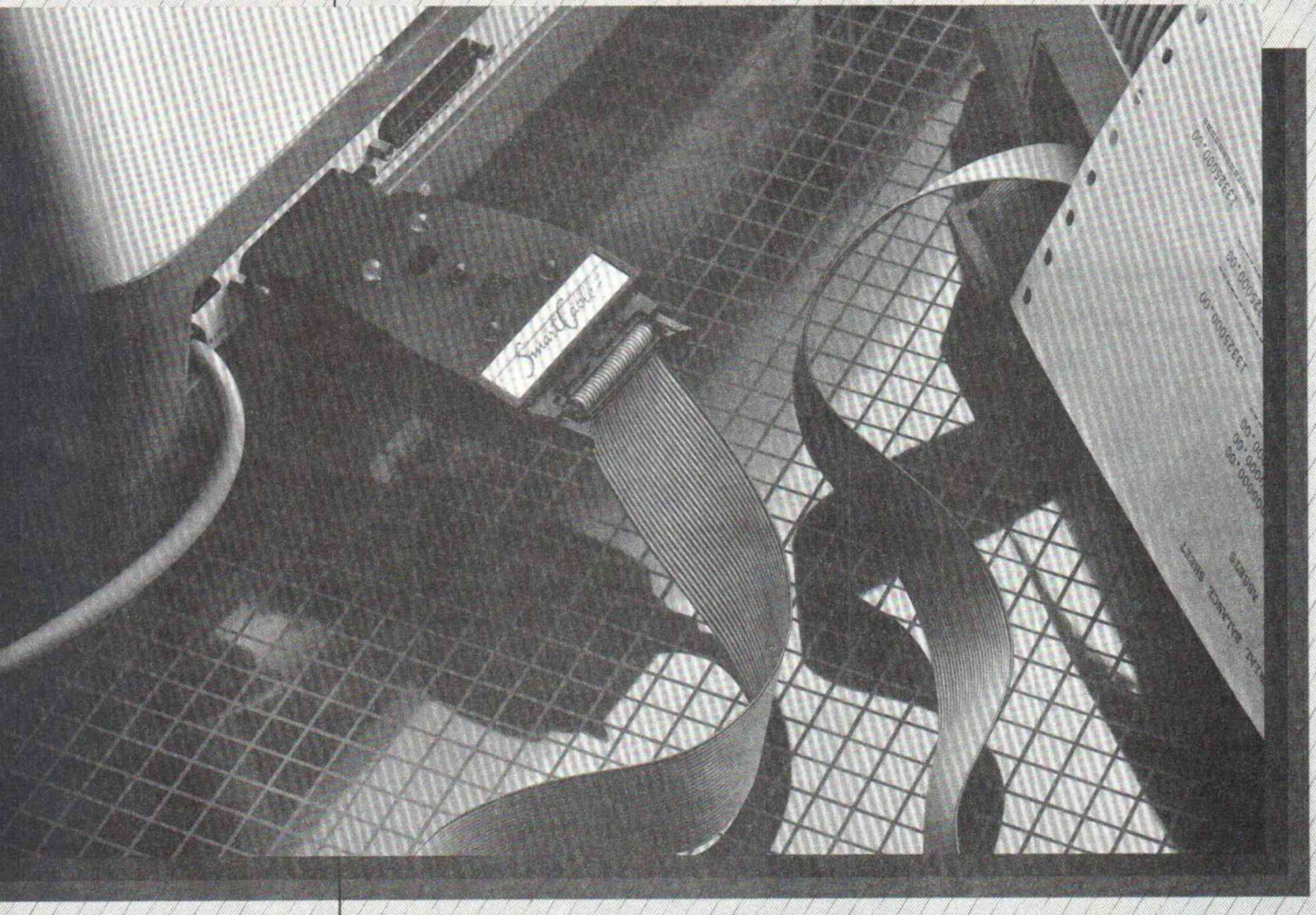
PORTABLE COMPUTER SUPPORT GROUP

1-214-351-0564

11035 Harry Hines Blvd., #207
Dallas, Texas 75229

TRAVELING COMPANION

The smartest way to connect your portable to the nearest printer.



You wanted your work to travel with you. That's why you bought a laptop computer. But until now, you couldn't *do* anything with that work until you got back to the printer or modem at the office.

Introducing the SC 817 Smart Cable from IQ Technologies. Your traveling computer's traveling companion.

Now you can connect to virtually anything. Anytime. Anywhere you go. The SC 817 is a universal RS-232 cable that uses its own internal logic to match virtually any computer to nearly any peripheral. Automatically.

In fact, it may be the only cable you'll ever need. On the road *or* at the office.

Give your dealer a call today. Because smart travelers take along the Smart Cable. From IQ Technologies.

IQ TECHNOLOGIES, INC.

11811 NE First Street, Bellevue, WA 98005
TELEX: 701 472 IQTECH UD (206) 451-0232

Circle 30 on Reader Service Card

In TELCOM, when Thinview is in use, the Prev key (F1) does not work. But because the screen is so much bigger, this is not a great loss. In BASIC and TELCOM, the label line at the bottom of the screen misbehaves. When toggled on and off by the LABEL key, the labels have a way of creeping up the screen with the scrolled text. One way to solve this problem is to avoid the LABEL key.

In TEXT, I did not encounter the problem of the spurious screen clearings. This is probably because TEXT does screen output differently than the other built-in programs, with lots of clear screen commands and direct addressing of the cursor. Thinview's problem with TEXT is that the reverse video (as during a cut or copy operation) does not work. The same deficiency shows up in editing of a BASIC program. A small box at the start of each line indicates which lines would have been in reverse video.

Thinview did not work at all with Lucid, but any BASIC program that works by repeatedly clearing the screen and repainting it should work fine.

THE HARDWARE

Thinview hooks up to the computer with a 2.5 foot flat cable. Nothing in the documentation indicates how the screen is to be positioned relative to the computer. Thinview can't really be put right on top of the built-in LCD since, even with SCREEN 1 selected, you must refer to the built in LCD every time you go to the main menu.

A makeshift typing stand comes with Thinview, and though no suggestions are given in the documentation on its use, one can prop Thinview up on the stand so it's above and behind the computer, oriented somewhat like the Tandy 200's pop-up screen. An optional mounting bracket is mentioned in the documentation. It presumably serves a similar function.

The LCD technology used in the Thinview results in characters that are harder to read than those of the Model 100, partly because of the smaller size (which can't be helped) and partly because the range of workable viewing angles is somewhat narrower. Characters are about 35 percent narrower than the Model 100 characters and about 40 percent shorter. Letters of the alphabet are shaped differently than on the Model 100 and values above 127 (e.g. generated by the CODE and GRPH keys) produce completely different characters on the Thinview. Additionally, the end-of-line marker seen in TEXT files is a small box rather than a triangle. However, just as on the 100, there is an adjustable polarization control to aid in optimizing the display for a given viewing angle.

One minor annoyance is that if one row of characters on the screen has many dots turned on (darkened), faint shadows will propagate above and below the characters. In extreme cases it almost looks like the characters rest on a mottled or striped backgound.

THE INSTALLATION

A 40-conductor flat cable runs from the Model 100 to the Thinview. The end that plugs into the Thinview (like the one at the computer end of the Model 100 printer cable) is a rather sturdy female connector. The other end is a DIP-style plug that plugs directly into the system bus of the Model 100. Axonix thoughtfully provides a zero insertion force (ZIF) socket to plug into the computer. The DIP plug may then be easily removed and replaced.

The DIP plug, though easy to remove, has very fragile pins. The prudent user should cushion it in styrofoam whenever it's out of the ZIF socket. For those who would repeatedly install and remove the cable, it's only a matter of time before pins will snap off. Therefore, you should expect to have to replace the DIP plug every so often.

Just as with the DVI, part of RAM must be used for an external-screen driver. Thinview takes up some 3,000 bytes for this function. A simple cassette loading operation sets up the driver and sets the RAM hooks to activate the BASIC keywords. Unfortunately, the Thinview software can give rise to some serious drawbacks for some users.

HOW THINVIEW HOOKS UP WITH ROM

When you're installing Thinview, it isn't enough just to hook up the hardware. You must also load some software. Two different problems can crop up as a result of the design.

The Thinview installation procedure calls for you to run a BASIC program from cassette. It protects memory from 60000 to MAXRAM (62959), loads a .CO file from tape into the now-protected area and calls an initialization routine at 60000. The protection of memory above 60000 is accomplished through the CLEAR command. The result is that HIMEM, a BASIC function, has the value of 60000.

The Model 100 ROM includes a number of hooks laying the way for installation of optional devices including DVI. The DVI hooks, each at a specific location in protected high memory (above MAXRAM) allow OPEN of the CRT device, setting of the CRT width and selection of the CRT as a default display device through the SCREEN command. (For a general discussion of hooks, see the August 1984 Portable 100.)

The hooks are located between FADAH and FB39H. The hook for opening the CRT, for example, is at FB1AH, and the hook for sending a character to the CRT is at FB1EH.

When you turn on a Model 100 for the first time, all the hooks, including the CRT hooks, are set to inactive values. For example, an attempt to OPEN the CRT generates a FC error. This is because the code at FB1AH is 08DBH. An attempt to execute the OPEN jumps to 08DBH, which is the routine that generates the BASIC function-call error message.

When Thinview software is installed, the initialization routine at 60000 changes the value at FB1AH (decimal 64282) to 60566. You can see this by typing PRINT PEEK(64282) + 256 * PEEK(64283) before and after Thinview software installation. In other words, after installation, when you try to OPEN the CRT, the ROM operating system, instead of generating an FC error, will jump to the machine-language routine at 60566.

The jump to 60566 lands in the middle of the protected RAM area (between 60000 and 62959) that had been loaded from the Thinview tape. A routine at 60566 performs the OPEN, and BASIC execution proceeds as usual.

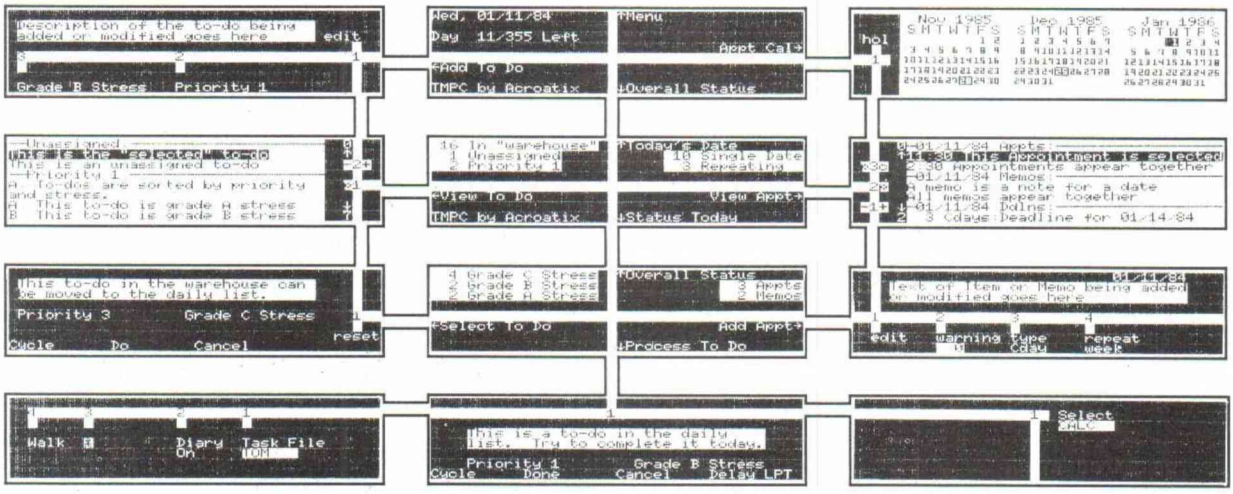
THE HOOK PROBLEM

But suppose the later execution of some other BASIC program includes statements that unprotect the area between 60000 and 62959. Or suppose the area remains protected but a different .CO file gets loaded into the area (from RAM or cassette). This is not at all unlikely. It occurs whenever Write Plus or the bar code reader is used.

The results are unpredictable and may result in a CPU-initiated cold start. Alternatively, the computer may lock up, forcing a manual cold start. One way to cold start it is by switching the memory power off. Another is to hold down the CTRL and BREAK keys and then push reset. The most visible effect of a cold start is that all user files are gone and the date resets to January 1,

TMPC

A Plan for Efficiency



For the TRS-80 Model 100

More than a Calendar

TMPC helps you organize your day. It is more than a calendar program; it is a toolbox to help you keep your most urgent tasks at top priority.

Efficiency is no Game

When you use TMPC, you enter the system of screens and corridors shown above and access features by moving through the rooms with the arrow keys. The screens may resemble an adventure game, but TMPC is a serious tool. The user interface will become second nature to you, like walking through the rooms in your home.

Functions for Efficiency

Each screen has its own function. For example, one screen includes a perpetual calendar, and another shows a "warehouse" of tasks to be done. In other screens you can set entries that repeat weekly, monthly, quarterly or yearly; request a warning of up to 999 days for any entry; sort your to-do list by stress level and priority; and more.

Organization Philosophy

TMPC has more than just "features;" it has a *philosophy* of time management inspired by Stephanie Winston's best seller, *The Organized Executive*. You won't learn all there is to know about TMPC in five minutes, but after using it, you'll feel more *organized*, not just more computerized.

Spreading the Word

One user told us that he felt *guilty* to get so much functionality at so low a price. Find out for yourself. Order TMPC today.

TMPC by Ac′rō-ă′tix
$49.95 + $3.00 Shipping

For more information, including how to get a *Free Sample*, call or Write:

Acroatix Incorporated
P.O. Box 273
Wilmington, MA 01887
(617) 658-5550

Phone orders only call

(800) 225-5800

VISA MasterCard

COD or Courier shipment available at extra charge.

Circle 2 on Reader Service Card

1900. Whenever a cold start occurs, the RAM hooks all return to inactive (default) values.

This homely anology may help to explain. When you move, it's often possible to have a recording installed on your old phone number telling people to call a new number. If you are at the new number, everything works out well.

Now suppose you are forcefully abducted from your new home. Someone trying to call you is refered to the new number. When they call, the phone is answered, not by you, but by an intruder. The caller will probably not be pleased by the results of the call.

The RAM hook value is like the recording referring the caller to the new number. Calling the new phone number is like jumping to the address specified at the hook. If some other .CO file has been loaded to the protected area, its code will be executed whenever one of the hooks is accessed. And it will be a rare surprise if the code there is even executable.

SOLVING THE HOOK PROBLEM

Thinview is not the only Model 100 product that uses RAM hooks and thus exposes you to the risk of unwanted cold starts. Another example is the Radio Shack bar code reader. But the bar code reader gives the user a way of cleaning up or reinitializing the hooks. The wand software, when installed, includes a routine for de-installing itself. (You de-install it by typing CALL 61807). Once it's gone, there is no danger of an unwanted cold start of the kind described above.

Version A of Thinview's software makes no provision for removing its RAM hooks. As a result, the only way to remove them is to store all user files elsewhere (e.g. on cassette), then cold start the machine and reload the files.

The user's manual for Version B includes an addendum giving a software routine for accomplishing the same end. It's a four-statement command that frees up the HIMEM-to-MAXRAM area and resets the RAM hooks. But the routine is cumbersome and easy to mistype. A simple CALL command, like that in the bar code software, would have been simpler.

THE .CO FILE PROBLEM

Machine language programs executed on the 80C85 processor (used in the Model 100) are assembled for execution in a particular location in memory. They cannot be moved to another part of memory and then executed. Because of the RAM organization of the Model 100, the usual place for a machine language program is just below MAXRAM and that's where Axonix puts the Thinview driver.

The Thinview 80-character by 25 line liquid crystal display system by Axonix Corporation of Salt Lake City, Utah.

Often one may wish to use more than one machine language program at a time. For example, while using Thinview you may wish to use an assembler, a bar code reader, Write Plus, Text Power 100 or Proaid, each of which also uses the machine language area just below MAXRAM. Some publishers of machine-language programs provide a relocating loader so you may put their software below other machine language programs. Axonix does not presently provide such a loader but says one is in development.

THE DOCUMENTATION

The Thinview manual, which lacks an index, isn't friendly to new users. The table of contents has only six entries. The checklist of package contents is incomplete, omitting, for example, the ZIF socket and the typing stand. Since it's so easy to destroy all user files when using Thinview there should have been prominent cautions at the beginning of the manual on the importance of backing up files. The instructions are imprecise on which way to orient the ZIF socket and don't explain at all what the software installation program is doing.

The manual discusses the BASIC commands OPEN FOR INPUT and OPEN FOR OUTPUT in connection with Thinview, but the Model 100 ROM makes it impossible to perform an OPEN FOR INPUT on the CRT device. There is no RAM hook for it, and an FC error will always result. Only the OPEN FOR OUTPUT command should have been discussed.

THE SUPPORT

Some companies operate with only a post office box and an answering service, and never return calls. Axonix, in contrast, prints an 800 number prominently in the manual, and has often given patient advice and counsel to me in my anonymous phone calls.

The first Thinview I received had certain bugs in the display driver software (version A). Some time later, I received the version B software reviewed here. Be sure to ask about the software version when placing an order.

In addition to the software problems, I had trouble with the hardware. The power switch isn't marked, and because of the automatic power-off design of the Thinview it is often very difficult for the new user to figure out which switch position is "on."

THE POSSIBILITIES

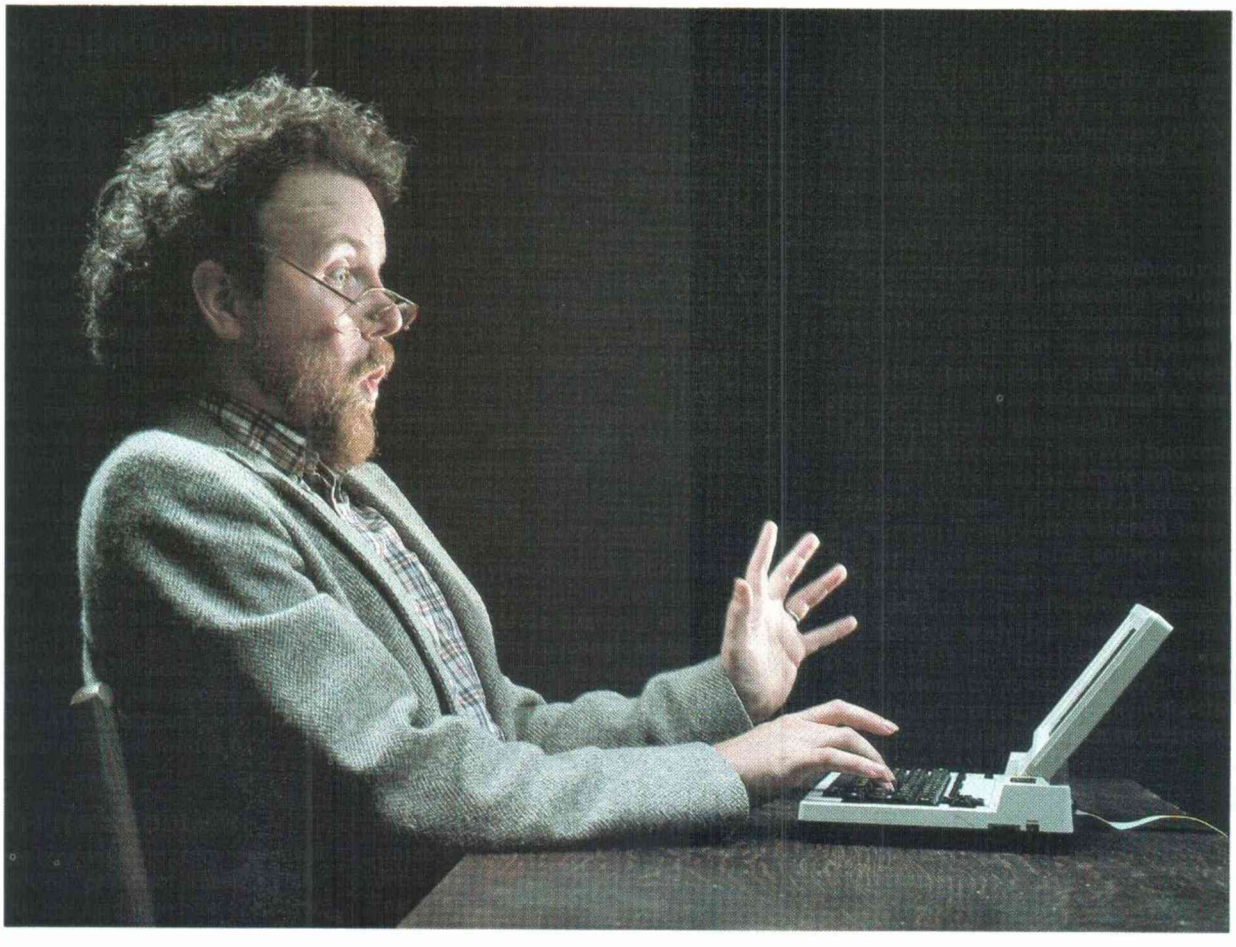
Axonix president Doug Kihm says a variety of options are now or will soon be available. These include memory modules, analog-to-digital interfaces and add-on Z80 processors.

Depending on the application and your budget, the Thinview may be just what you need. □

Good? Yes. Ultimate? Well, Maybe.

A neat little three-in-one package that will blow you away. Ultimate ROM brings together Traveling Software's Idea, T-Writer and T-Base all on one 32K chip. It's faster, easier and more efficient than its cassette-based counterparts.

BY CARL OPPEDAHL AND J.D. HILDEBRAND



PHOTOGRAPHY BY BENJAMIN MAGRO

Some purchasers of Seattle-based Traveling Software's Ultimate ROM received an unexpected bonus — a 65-cent Marvel comic book detailing the exploits of a robot-like superhero named (appropriately enough) ROM.

With typewriter correction fluid and a blue ball-point pen, someone at Traveling Software's promotion department customized certain pages. "Reach for the stars with *the Ultimate ROM!*" the doctored comic exhorts. "Join the young *ROM* program and face the challenge of *lap computing*. You can take part in exciting *ROM*-related educational activities. . . ."

The comic book is more than a high-tech Mad-Libs game. It confirms, for instance, Traveling Software's role as the industry's most lighthearted software publisher — a position in some jeopardy since the Traveling Professor's photographs were replaced by pen-and-ink drawings. (Someone started a rumor at a recent trade show that the professor had defected from Traveling Software and gone to work for a competitor in the laptop market. "Not true," says Traveling Software president Mark Eppley. "The professor is on sabbatical. He's developing new products and playing a lot of golf.")

More importantly, the promotion serves to highlight the significance of the Ultimate ROM for Traveling Software and laptop users.

ULTIMATE QUESTIONS

One of the 100's major limitations is having enough space for storing programs and text files. Once you load a few third-party packages and save a TEXT file or three, you'll notice your valuable RAM getting soaked up like spilled coffee in a paper towel commercial. What to do?

There's an obvious answer: You can add more RAM. But that's a finite alternative that only delays the inevitable.

There's an expensive answer: You could add a disk drive or tape back-up for storing files and programs that aren't being used. But these tend to detract from your computer's overall portability.

And there's the answer Traveling Software calls "ultimate": You can buy programs stored on read-only memory (ROM) chips. This alternative provides a happy medium because it increases the amount of RAM available for storing files without compromising portability.

This solution provides additional benefits:

- It makes program-loading as simple as plugging in a chip — say goodbye to awkward cables and bulky tape drives.
- It protects programs from accidental modification or deletion.
- It loads more quickly and runs more efficiently than cassette-based alternatives.
- It makes applications programs available anytime, no matter where users are or how far they are from their cassette drives.
- Since the programs reside in ROM and consume virtually no RAM, data files can be much larger.

Traveling Software isn't the first com-

(continued on page 64)

Ultimate ROM
ROM-based software
Traveling Software
11050 Fifth Ave. N.E.
Seattle, WA 98125
(206) 367-8090
Requires: Model 100, Tandy 200 or NEC PC-8201
$229.85
Circle No. 141

A NEW MARKET FOR LAPTOPS

Laptop machines like the Model 100 and Tandy 200 are already runaway favorites with journalists, hobbyists, students and others. An enormous new corporate market is just around the corner, thanks to the firmware methods exemplified in Ultimate ROM. I'm talking about corporations that require a few hundred or a few thousand customized computers to solve some sort of recurring problem. Here are a few examples:

- A life insurance company wants its sales force of several thousand agents to be able to quote premiums and cash values for policies without having to thumb through books of charts and tables.
- A maker of photocopiers wants its field service technicians to be able to run diagostics on microcomputer-controlled copiers through a serial port.
- A grocery store chain wants to design a system allowing store personnel to order new stock by walking down the aisles of the store and scanning bar code stickers on the shelves, then transmitting the order by phone.

Each of these applications could be solved with a laptop machine and a custom-made option ROM. These solutions are far less expensive, and more user-oriented, than any other technology.

Consider the insurance sales example. Experiments equipping salespeople with luggable machines such as Kaypro and Compaq have generally failed. The machines are heavy (10 times heavier than a Model 100), require you to fumble around for an electrical outlet and rely on comparatively fragile disk drives. Even recent battery-powered disk-based laptops such as the Kaypro 2000 and the Data General One have problems — mainly they're heavy and about 10 times the cost of Kyocera machines.

In copier maintenance, the traditional field service aid is a custom-designed and hand-assembled piece of hardware, a test adapter. The development and manufacturing cost, amortized over only a few hundred adapters in the whole country, is almost $100,000 per adapter. In contrast, a Tandy 200 with a custom ROM may have a per-unit cost in the low thousands. The test program, written in powerful, interrupt-driven Microsoft BASIC, requires minimal development hardware and is easily debugged. A full LCD screen is available, compared to the usual two or three digit LED display on an adapter.

The grocery store ordering device I'm thinking of stores its order information on a cassette tape and transmits it later over phone lines. Each unit costs about $10,000 — and much more in repair costs. There is no editing capability, so if you scanned "raisin bran" by mistake there's no way to prevent having a case shipped to you with the rest of your order. Think how handy an LCD screen and a goof-proof ROM program would be.

If you custom design and build a few hundred pieces of hardware to solve a problem, it is going to be very difficult to modify the equipment later. On the other hand, with a laptop machine, it is easy to pop out one ROM and pop in another. And how will you service the custom hardware when it breaks? The laptop can be easily replaced or repaired by making a trip to the corner Radio Shack store.

Entrepreneurs: Take note!

—Carl Oppedahl

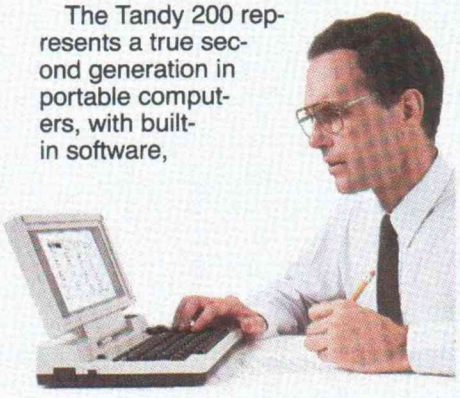
AT LAST, WORD PROCESSING THAT WON'T COME UP SHORT.

The Tandy 200 communicates ideas better.

Our Tandy 200 portable computer does what no other computer could. It takes our amazing Model 100 one step further. And like the Model 100, the powerful Tandy 200 portable computer includes state-of-the-art features and performance at an unmatched price.

The Next Generation of Portable Computing

The Tandy 200 represents a true second generation in portable computers, with built-in software,



improved features, a bigger screen and a larger memory. All designed to make the Tandy 200 a more potent portable problem-solver.

For complex spreadsheet analysis and calculations, we put popular Multiplan software into the Tandy 200's permanent memory. Just turn on your portable Tandy 200, press a button and you've got a powerful decision-support tool at your fingertips. Now it's easier than ever to do accurate sales forecasts, profit and loss projections, budgeting, pricing, engineering calculations and more.

Improved Word Processing Software

An improved version of the Model 100's easy-to-use word processing program makes the Tandy 200 especially useful for journalists, salespeople, engineers, students and anyone who needs to write letter-perfect memos, reports and correspondence in a hurry. Edit, delete and move blocks of text with just the touch of a function key. It's that easy. The Tandy 200 gives you the convenience of a personal word processor for use anytime, anywhere for notes.

Four other "instant-on" programs let you use the Tandy 200 as your personal appointment calendar, address and phone directory, and telephone auto-dialer (the Tandy 200 generates tone dialing pulses, so you can use it with long-distance services). A much more powerful built-in program for communications makes it easy to access other computers by phone. You can even create your own programs in the BASIC language, too.

A Bigger Screen and a Larger Memory

The Tandy 200's high-contrast flip-up liquid crystal display has 240 × 128 resolution for big, clear graphics and easy-to-read characters. The 16-line by 40-character display makes the Tandy 200 particularly well suited to word processing and spreadsheet duties and with simple BASIC commands you can add dot-addressable graphics to your text.

To get the most from the Tandy 200's powerful new capabilities, we increased the standard memory size to more than 24,000 characters. Additional banks of 24K modules let you increase the memory up to 72K.

The Best in Technology for Under $1000

You can start with the powerful software and impressive capabilities of the Tandy 200 (26-3860) for just $999. Best of all, the Tandy 200 has parallel printer, RS-232C, cassette and bar code reader interfaces for added versatility. You can even add disk storage and a monitor for a complete desktop system. Get a hands-on demonstration at Radio Shack today.

Prices apply at Radio Shack Computer Centers and at participating stores and dealers.

Radio Shack®
The Technology Store™
A DIVISION OF TANDY CORPORATION

Circle 47 on Reader Service Card

New 1986 Computer Catalog. Send me a copy.
Mail To: Radio Shack, Dept. 86-A-230, 300 One Tandy Center, Fort Worth, TX 76102

NAME _____
COMPANY _____
ADDRESS _____
CITY _____ STATE _____ ZIP _____
PHONE _____

What People Are Saying About Traveling Software®...

For TRS-80 Model 100, NEC PC-8201, and Olivetti M10 laptop computers:

IDEA™

"This thing is fantastic. I can write proposals in about 50% of the time it took previously, I can organize presentations better — I use it for just about everything." Jonathan W. Pearce, C.P.A.

"I found it an invaluable tool. It has become as much a part of my 100 as Text and Telcom, like a second skin." John P. Mello, Portable 100/200, March, 1985

Organize anything from simple lists to complex outlines. Easily reorganize and search. Use word processing and database capabilities. Develop outlines from rough drafts into complete papers. $79.95

T-backup™

"Your program is super, and a real time saver." Bruce Arnold, special sales manager

"Your T-backup program is the most useful that I have ever purchased for my Model 100. Keep up the good work." Thomas Fehsenfeld, customer.

Save the entire contents to cassette from RAM in minutes, rather than back up files individually. Check the integrity of saved data copies. Restore data from cassette. $19.95

T-base™

"Cramming the complexity of a relational database manager into a portable computer ... may seem ambitious ... Yet Traveling Software's T-base handles the challenge with surprising grace..." P. Gregory Springer, InfoWorld, December 10, 1984

Design relational data bases. Set up screen files. Borrow information from fields in other data bases. Perform math computations. Create custom-printed or LCD display reports. $99.95

T-Writer™

"...all that it is advertised ... superior in many aspects to the WRITE+ program that I used previously." John J. Atkins Jr., food service director.

Print documents created using the built-in TEXT program. Include justification, headings, footings, underlining, italics, boldface, and more. Produce form letters and mailing labels. $49.95

Traveling Appointment Manager™

"The program and manual were user friendly ... an excellent value." Mary Jeann Batham, Portable 100/200, May 1984

Maintain a detailed list of appointments and things-to-do. Consult on-screen or printed records. Automatically reschedule recurring appointments. $59.95

Traveling Expense Manager™

"...the Traveling Expense Manager is well-thought-out and keeps track of exactly the type of information that you need while you are on the road. It's everything you need to keep your boss — and the IRS — happy." Dan Robinson, InfoWorld, July 30, 1984

Define expense categories. Record expenses to any account. Indicate if a business expense is billable. Produce reports by client or project, and the type of expense. $59.95

Traveling Project Manager™

"...very useful in helping me track my projects ... simple ... provides useful data and printouts." J. Michael Masterson, strategic projects manager

Simultaneously manage several projects — small or large. Enter dollar or other unit expenditures. Produce a variety of summaries and printed reports. $59.95

Traveling Sales Manager™

"I spend a lot of time on the road and find the Traveling Sales Manager useful... The manual, in particular, was outstanding." Don Gillespie, salesman

Keep a detailed log of customers and contacts. Record what customers purchase and when. Store important biographical data and reminders to call on customers. $59.95

Traveling Time Manager™

"...first-class software, a valuable tool for consultants and others billing clients by the hour." John P. Mello, Portable 100/200, April 1984

Track time spent on projects or for a client. Apply your own billing rate. Indentify time expenditures as billable or nonbillable. $59.95

To order, or find the dealer nearest you,

CALL 1-800-343-8080

For information or technical assistance,

CALL (206) 367-8090

TRAVELING SOFTWARE®

Traveling Software, Inc.
11050 Fifth Avenue, N.E
Seattle, WA 98125
(206) 367-8090

Turbo charge Your BASIC Programs

By Dale Flanagan

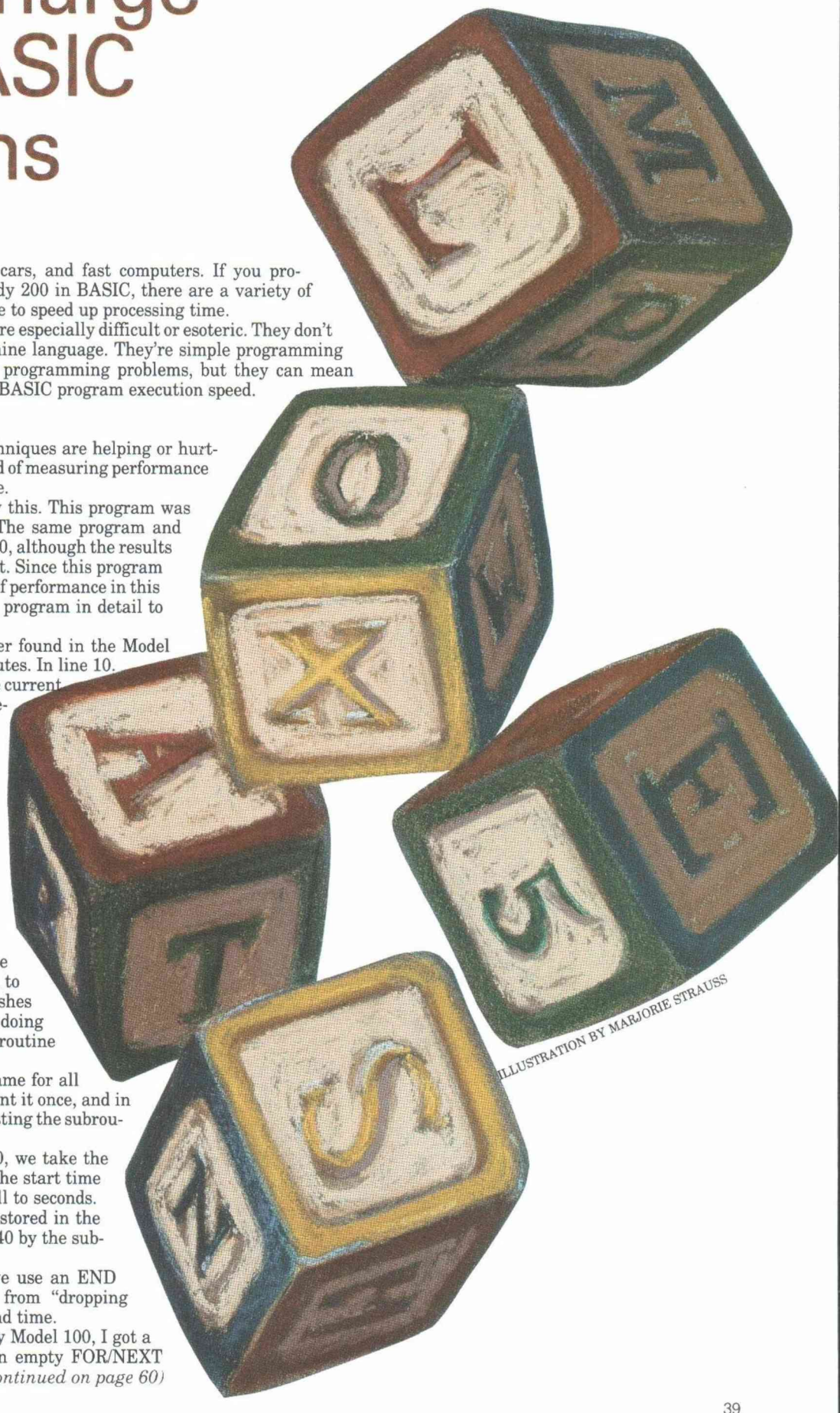


ILLUSTRATION BY MARJORIE STRAUSS

Americans like fast food, fast cars, and fast computers. If you program your Model 100 or Tandy 200 in BASIC, there are a variety of simple techniques you can use to speed up processing time.

None of these techniques are especially difficult or esoteric. They don't involve any tricks or the use of machine language. They're simple programming techniques you can use in everyday programming problems, but they can mean significant improvements in 100/200 BASIC program execution speed.

A BASIC BASELINE

To find out what programming techniques are helping or hurting, we have to first establish a method of measuring performance and a baseline from which to measure.

Listing 1 — TEST1 — does exactly this. This program was written and run on my Model 100. The same program and techniques can be used on your 100/200, although the results you receive might be slightly different. Since this program forms the basis for our measurement of performance in this article, it's well worth looking at the program in detail to see what it's doing.

The program uses the built-in timer found in the Model 100 to track how fast a program executes. In line 10, a variable (START$) is set equal to the current time (found in TIME$). TIME$ is a special variable pre-defined on the Model 100, and it's constantly changing as the computer's internal clock ticks along. Because of this, we have to preserve the starting time if we want to calculate how long it takes for a program to execute.

Lines 20 and 30 are simply an empty FOR/NEXT loop. The program is in "idle" at this point, simply killing time. The FOR/NEXT loop gives us a baseline to measure the effects of changes in the program. In line 40 we set another variable, FINISH$, to the time found when the program finishes the empty FOR/NEXT loop. After doing this, the program goes through a subroutine starting at line 500.

Since this subroutine will be the same for all examples in this article, we'll only print it once, and in subsequent listings we won't bother listing the subroutine.

In the subroutine found at line 500, we take the hour, minutes, and seconds between the start time and the end time and convert them all to seconds. This total elapsed time in seconds is stored in the variable "T" and printed out in line 540 by the subroutine.

In line 499 of the test program, we use an END statement to prevent the program from "dropping through" to the subroutine for a second time.

When I ran this test program on my Model 100, I got a consistent result of 15 seconds for an empty FOR/NEXT

(continued on page 60)

An Orchestra in Your 100

By Chris Shaw

I purchased an 8K Model 100 for making field notes and transferring them later to a desktop computer. Reading the manual, I discovered the 100 has a tone generator, and I thought it might be enjoyable to use it for capturing musical notes — tunes that sometimes pop into my head that I'd like to remember for the symphony or concerto I'll write when I retire.

The first thing to do when building any program is to decide on requirements — and it helps to keep them simple. My particular requirements for capturing music were to record short tunes, including the tone and duration of each note, using the 100's keyboard as a real-time piano keyboard; to play back these recorded tunes; to have a simple user interface; to be similar in operation to the 100's TEXT; and to store the recorded tunes as text files, so they can be edited by TEXT.

Could the 100 meet these requirements? In particular, could it record, in real-time, both the tone and duration of each note?

A closer study of the manual yielded the information that the only way to interrupt the SOUND command, in BASIC, is by defining function key interrupts, with the ON KEY GOSUB command. Since the eight tiny function keys on the 100 make an even worse piano keyboard than the letter keys, I needed another solution. When keying in a tune, could I represent each note as a rapidly-repeated series of very short tones, and check for a keyboard input between each one?

I wrote this one-line BASIC program to find out:

```
10 SOUND 5000,2: K$ = INKEY$: IF K$ = "" THEN 10
```

This produced a somewhat annoying sound, but I could control its duration with the keyboard, so it should work. (Sometimes it's a good idea to write some code even before the system is designed.)

Having satisfied myself on the feasibility of the project, my next step was to design the user interface.

THE CREATIVITY FACTOR

Designing a good user interface takes a certain amount of creativity. After considering a number of possibilities, I decided on this set of commands:

Load: To operate like TEXT, my program, MUSIC, prompts for a file name, and loads the notes from the file into an array in the program. If there's no file, MUSIC starts with an empty note array.

Play: Press the ENTER key to play all

(continued on page 48)

GRAPHIC BY ELIZABETH MARITATO

PORTABLE BASICS

For your Model 100, TANDY 200,

or NEC 8201

UTILITY PROGRAMS: *Everyone needs at least one of these programs. THEY ARE AN EXCEPTIONAL COMPUTING VALUE. Written in BASIC — No conflict with Machine Language programs, DVI, or other peripherals. Only $14.95 for your first program, $9.95 for each additional one (in the same order). Buy 3 and choose a 4th program free!*

RENUMBER — *2K — Renumbers Basic. Programs in place. (Models 100 and 200 only).*

DISASSEMBLER — *1.3K — Perhaps the smallest disassembler around. Output to LCD, LPT, RAM, COM, or CAS.*

ASSEMBLER — *2.2K — Compile your own machine language programs. Slow, but effective. Approx. 30 bytes/minute.*

CONVERT — *.5K — Conversions between Decimal, Hex, and Binary numbers.*

HEXSAVE — *.5K — Save and Reload Machine Language programs using ASCII data. Allows uploading to your desktop computer.*

ANNUITY CALCULATOR — *1.5K — Present Value, Future Value, Payments, Interest Rate. Compare loans or plan for the future!*

CALENDAR — *1K — Display any month since 1582 AD! Scroll forward, backward, or enter new month with one key.*

TIMER/ALARM — *1 K — Countdown timer, stopwatch, or alarm. Perfect for eliminating missed wake-up calls!*

FILE SIZE — *.6K — Display the RAM address and number of bytes used by each of your files. Helps with memory management.*

LINE FEED — *.3K — Enable/Disable Line feeds instead of flipping DIP switches. Status displayed or changed from main menu. (Models 100 and 200 only).*

SCREEN DUMP — *.5K — Send Graphics to your printer! Most Daisywheel and Dot Matrix printers can be used. Send us your printer control codes so we can set up the program for you.*

STATISTICS — *1K — Compute Mean, Standard Deviation, and Variance of a Normal distribution. Reads data from a text file.*

WEIBULL STATISTICS — *1K — Characteristic Life, Reliability, and Confidence. Multiple Samples, Suspended Tests.*

MACHINE LANGUAGE PROGRAMS: *$17.95 each. All of them are Relocatable! And none of them need any free RAM. None.*

CASSETTE — *2K — Back up all your files at once. Only takes 4 minutes for a 32K save or load. (Models 100 and 200 only).*

SCREEN DUMP — *.5K — Similar to the BASIC version, but much faster. Send us your printer codes when ordering this.*

ASSEMBLER — *1.4K — Includes ORG, DATA, STR, RES, EQU, and $ pseudo-ops. Compact and FAST!*

Need something that isn't here? **CUSTOM** *Software services available, too. Send me your application, I'll send you a quote. Most jobs $50–$250.*

These are good, well-written programs. Don't let their size fool you. A lot of work was done to insure minimum use of your portable's RAM. The prices are low because the author and seller are the same person. That also means excellent customer support!

ORDERING INFO:

Add $2.00 S&H per order. Michigan Residents add 4% sales tax. Checks, MasterCard, and VISA accepted. No Purchase Orders unless payment is included. No COD's. Thank You very much!

PORTABLE BASICS
P.O. Box 696
Comstock, MI 49041
616-343-9315
CompuServe I.D. #: 72346,1020

Introducing Portable Basics! We are a new company serving the everyday needs of the PoCo market. We won't be selling ROMs or 10K database programs. We will be selling programs consuming no more than 3K. They will be well-written, easy to use, and inexpensive. Many programs similar to ours are available on Compuserve or in magazine listings. Our programs are typically easier to use, smaller, and have better documentation. Also, Portable Basics eliminates downloading and typing problems! We want Portable Basics to be shaped by you, our customers. We welcome your input for future products. Send us a letter or give us a call. We'd like to hear from you! Sincerely, Jim and Jan Irwin

05-Nov-85 04:09:54
Sb: P100/200 correction
Fm: Mike Nugent 71426,1201

... a couple of errors in my UNKEY$ article (Oct. '85 P100/200).
The last part of line 210 should read POKE HIMEM+17,245
  That's all! Thanks! -- Nuge

UNKEY$: Selective Amnesia for the Model 100 Keyboard

If at first you don't succeed, write a program to get around it. Here's how one programmer beat the system at its own game.

By Michael J. Nugent Jr.

Stupidity is the mother of invention. Well, okay — necessity is the mother of invention, but sometimes stupidity is what makes it necessary. I did something dumb and had to invent a way around it. Let me explain.

Back in my Model III days I prevented novice users from halting program execution by disabling the computer's Break key — all it required was a couple of POKE statements. More recently, after writing and debugging a long, complicated Model 100 program, I added the equivalent statement: POKE 63056,128. That's when I discovered something I should have checked out before writing my program — this POKE has some side effects.

Unfortunately, the command disables the function keys; I hadn't counted on that. And although Break/Pause is blocked, it isn't gone. The keystrokes are still stored and will assert themselves when the block is removed by POKEing 63056,0 or by using the INPUT, LINE INPUT, or INPUT$ statements — INKEY$ becomes the only safe way to get keyboard input. Finally, the Print key is not disabled. It's still possible to lock up the system in an I/O loop by pressing Print with the printer off-line. Then, while the system waits for the printer, pressing Shift-Break or Ctrl-C will halt the program.

There had to be a better way. With some help from Christopher Morgan's *Hidden Powers of the TRS-80 Model 100* I devised a utility I call UNKEY$. With UNKEY$ resident in high memory, a simple POKE completely disables the Break/Pause, Ctrl-C, Ctrl-S, and Print keys. You can turn UNKEY$ on and off at will throughout your programs without fear of stored keystrokes coming back to haunt you. INPUT, LINE INPUT, and INPUT$ statements can be used, and the function keys remain fully operational.

As a bonus, UNKEY$ corrects the Model 100's DATE$ bug.

HOUSEKEEPING BUFFERS

The Model 100 does many housekeeping tasks in the background, using the processor's clock to trigger the necessary actions. One such task is to scan the keyboard and decode any keystrokes.

For this purpose, two nine-byte buffers in random-access memory (RAM) store images of the keyboard matrix. Let's give them some clever names: BUFFER1 and BUFFER2.

BUFFER1's job is to remember which keys were pressed during the most recent scan. Its nine bytes present a complete image of the keyboard, with *1* bits to indicate keys which were pressed and *0* bits for the others.

When a scan detects a pressed key, it's recorded in BUFFER1. At the same time, the system checks to see if that key was already in BUFFER1, meaning that it had been detected on the previous scan too. If so, then the key has been "officially recognized" by the system and is copied into BUFFER2, where appropriate action will follow.

To be recognized by the system, a key must be detected on two consecutive scans. Aha! What if, between scans, we could make BUFFER1 "forget" certain keys? That's what UNKEY$ does. Here's how.

The Model 100's designers left us a hook at F5FFh — a RET, followed by two NOPs. On each "tick" of the clock, the system calls F5FFh, returns immediately and continues on its way. We'll replace that hook with a JMP to our own program, forcing a little detour through UNKEY$ every time the clock ticks.

Since the system scans the keyboard only every third tick, according to a counter at FF8Fh, that's when we'll fix BUFFER1. (We can still make use of the other ticks.) So our first task is to check the counter to determine the current tick.

On the first tick UNKEY$ just exits back to the system. On the second tick UNKEY$ takes a moment to deal with the infamous DATE$ bug (as long as we're in the neighborhood) by POKEing the proper year into locations F92Dh and F92Eh. On the third tick the fun begins — maybe. Is the disable flag set? If not, UNKEY$ grudgingly exits. Shucks!

PURGING BUFFER1

Otherwise, it's time to rid BUFFER1 of the offending keys. First, we'll "forget" the Print key at FF97h by ANDing that byte with BFh, forcing bit 6 to zero. Next, a quick shot of amnesia to bit 7 of FF99h, and Break/Pause is history. Keeping Ctrl-C and Ctrl-S off of the final exam requires a little IF/THEN stuff. We test bit 1 of FF99h to see if Ctrl is pressed. If it is, then we un-remember the "C" and "S" keys in the same fashion as above. BUFFER1 has just had a memory lapse. Exit (smiling) to system.

Once UNKEY$ is installed it needs no maintenance except after a cold boot. By hooking into the background task, UNKEY$ becomes part of the operating system, working all the time. Just put the appropriate POKE statements in your BASIC programs when you want to disable and re-enable the keys.

AMNESIA

Listing One: LOADER.BA gives the Model 100 direct keyboard control.

```
10  READ SIZ
20  NH=HIMEM-SIZ
30  CLS
    :PRINT "New HIMEM =" NH
40  CLEAR 256,NH
50  READ SIZ
60  FOR PTR=0 TO SIZE-1
70  ADR=HIMEM+PTR
80  READ A
90  IF A=>0 THEN 150
100 READ A,B
110 REL=A+256*B
120 FX=REL+HIMEM
130 B=INT(FX/256)
    :A=FX-B*256
140 POKE ADR+1,B
    :PTR=PTR+1
150 POKE ADR,A
160 NEXT
170 OLDVEC=HIMEM+75
180 FOR X=0 TO 2
    :POKE OLDVEC+X,PEEK (62975+X)
    :NEXT
190 BEEP
    :PRINT "Are you using a bar code reader?";
    :I$=INPUT$(1)
    :PRINT
200 I=INSTR("YyNn",I$)
    :IF I=0 THEN 190
210 IF I<3 THEN FLAG=HIMEM+SIZ-3 ELSE FLAG=62971
    :POKE HIMEM+16,251
    :POKE HIMEM+17,24
220 B=INT((HIMEM+3)/256)
    :A=(HIMEM+3)-B*256
230 POKE 62975,201
240 POKE 62977,B
    :POKE 62976,A
250 POKE 62975,195
260 PRINT "UNKEY$.CO is now installed..."
270 PRINT "FLAG =" FLAG
280 PRINT "POKE " STR$(FLAG) ",1 to disable keys"
290 PRINT "POKE " STR$(FLAG) ",0 to re-enable keys
300 END
60010 DATA 81
60020 DATA 77,74,78,245,58,143,255,254
60030 DATA 2,250,-1,74,0,202,-1,62
60040 DATA 0,58,-1,78,0,183,202,-1
60050 DATA 74,0,58,151,255,230,191,50
60060 DATA 151,255,58,153,255,230,127,50
60070 DATA 153,255,230,2,202,-1,74,0
60080 DATA 58,145,255,230,251,50,145,255
60090 DATA 58,146,255,230,253,50,146,255
60100 DATA 195,-1,74,0,58,-1,79,0
```

```
6Ø11Ø DATA 5Ø,46,249,58,-1,8Ø,Ø,5Ø
6Ø12Ø DATA 45,249,241,2Ø1,Ø,Ø,Ø
6Ø13Ø DATA 8,5
```

Listing Two: Remove.BA searches for the UNKEY$ routine and removes it entirely.

```
1Ø  CLEAR 256,HIMEM
    :CLS
2Ø  A=PEEK (62976)+256*PEEK (62977)-3
3Ø  IF CHR$(PEEK (A))+CHR$(PEEK (A+1))+
    CHR$(PEEK (A+2))="MJN" THEN 5Ø
4Ø  BEEP
    :PRINT @12Ø,"UNKEY$ not found or inactive."
    :END
5Ø  OLDVEC=75
    :SIZE=81
6Ø  POKE 62975,2Ø1
7Ø  FOR I=2 TO Ø STEP -1
8Ø  POKE 62975+I,PEEK (A+OLDVEC+I)
9Ø  NEXT
1ØØ C=-SIZE*(A=HIMEM)
11Ø BEEP
    :PRINT @12Ø,C "bytes freed. HIMEM =" HIMEM
12Ø CLEAR 256,HIMEM+C
13Ø END
```

Table One: Memory addresses for the Model 100 keyboard.

BUFFER1 AND BUFFER2

Buffer1	7	6	5	4	3	2	1	Ø	Buffer2
FF91h									FF9Ah
65425d	L	M	N	B	V	C	X	Z	65434d
FF92h									FF9Bh
65426d	K	J	H	G	F	D	S	A	65435d
FF93h									FF9Ch
65427d	I	U	Y	T	R	E	W	Q	65436d
FF94h	?	>	<	"	:	rbrkt			FF9Dh
65428d	/	.	,	'	;	lbrkt	P	O	65437d
FF95h	*	&	caret	%	$	#	@	!	FF9Eh
65429d	8	7	6	5	4	3	2	1	65438d
FF96h					+	unlin	)	(	FF9Fh
65430d	darw	uarw	rarw	larw	=	-	Ø	9	65439d
FF97h							del		FFAØh
65431d	entr	prnt	labl	paste	esc	tab	bksp	space	6544Ød
FF98h									FFA1h
65432d	f8	f7	f6	f5	f4	f3	f2	f1	65441d
FF99h	brk	not							FFA2h
65433d	pause	used	cap	num	code	grph	ctrl	shft	65442d

AMNESIA

INSTALLATION & DEMOLITION

To avoid disturbing any programs already in high memory, LOADER.BA installs UNKEY$ just below HIMEM and then resets HIMEM. Just in case another program already uses the hook at F5FFh, the loader copies the three bytes of the hook and uses them as UNKEY$'s exit. Thus, after a detour through UNKEY$, the system will continue toward its original destination.

LOADER also determines the POKE address for the disable flag. If you use a bar code reader, the loader uses the address of a flag byte in UNKEY$, which varies according to where UNKEY$ resides in memory. If you don't use a bar code reader, the loader uses 62971 as the flag byte, which never varies, regardless of where UNKEY$ is installed. LOADER will tell you the POKE address for your particular computer.

If for any reason you should want to deactivate UNKEY$ completely, use REMOVE.BA. This program finds UNKEY$ in memory and restores the original contents of the hook, removing our detour. Then, if there are no programs loaded below UNKEY$, it resets HIMEM, giving the free bytes back to BASIC.

A WORD OF WARNING

Since UNKEY$ acts as part of the operating system, keys are always disabled while the flag is set — not just while in BASIC. So before leaving BASIC, you may want to be sure the flag byte is zero. If you forget to do this, you can just re-enter BASIC and do the POKE. Aside from the POKE, the only other way to re-enable the keys is to do a cold boot.

You must avoid letting a program overwrite UNKEY$, or the system will jump to who-knows-what and may never return. Programs that CLEAR xxx, MAXRAM (or some value above HIMEM) leave UNKEY$ unprotected. Either modify those programs to respect HIMEM, or run REMOVE.BA before running the offending programs and LOADER.BA afterward to reinstall UNKEY$.

It's also not a good idea to install UNKEY$ a second time without first having run REMOVE.BA to delete the existing copy of UNKEY$. It not only wastes memory, but under some circumstances, might be fatal to your files.

WANT A DATE, SAILOR?

Okay, just a few more points to cover, and then I'll leave you alone so you can go play with a spreadsheet.

AMNESIA

Each year, you can modify the DATE$ year without stopping UNKEY$. We'll use 1986 as an example. From BASIC, enter the following commands:

```
YR = PEEK(62976) + 256 * PEEK
  (62977) + 76
POKE YR,8
POKE YR+1,6
PRINT DATE$
```

To update your copy of LOADER.BA to reflect the new year, change the DATA statement in line 60130 and re-save it.

PEACEFUL COEXISTENCE

UNKEY$ does what it can to avoid stomping on other programs you may be using. Sadly, not all programs take such precautions, and since I use few commercial programs, I have no way of telling which ones will coexist peacefully with UNKEY$. Be sure to back up your files before trying any new software.

When using UNKEY$ as part of a password protection scheme, you may want to take extra steps to ensure security. Use the IPL command to force a warm start to run your password program. Use ON ERROR GOTO so bad input data doesn't cause errors to halt your program.

UNKEY$ is a simple and useful little utility, but it's only the beginning. We've learned how to intercept the keyboard scan and capture keystrokes. Having kidnapped the keystrokes, we can do anything we choose. We could enable rather than disable keys, say, forcing all input to upper case by making the machine think the Caps-lock key is depressed.

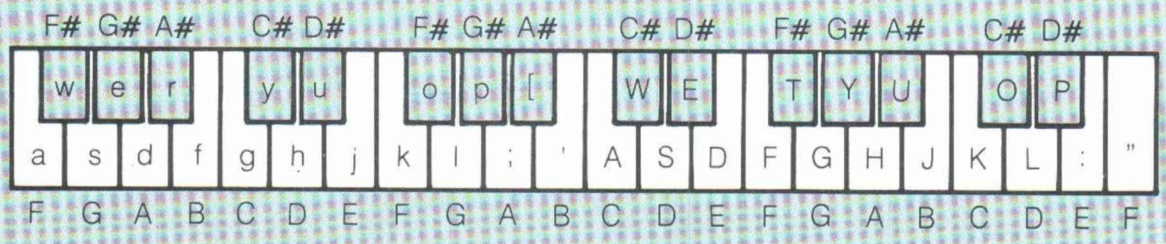
Sometimes the 100 comes out of the briefcase with the NUM key locked down. If you don't notice it and type LIST you're actually typing 35ST — inserting a nonsensical line 35 into your program, blowing away its previous contents. If the NUM key were disabled, this couldn't happen.

Want to disable the arrow keys? Or enable *only* the arrow keys? Change a key into another key?

I'm working on some similar goodies, which I'll gladly share with you — unless I decide to become a rock star, in which case you're on your own. In the meantime address your questions, comments or special requests to me via Portable 100/200. □

Please help us rate this article's value. If you've found it very valuable, circle 164 on the Reader Service Card. If it was moderately valuable, circle 165—and if it wasn't valuable to you, circle 166.

Figure 1



the notes currently stored in the array.

Backspace: Press the BKSP key to delete the last note stored in the array.

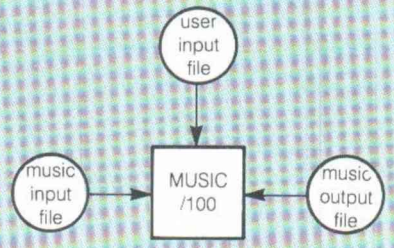
Erase: Press the DEL key (SHIFT BKSP) to delete all notes in the array.

Save: Press the F8 function key to save all notes in the array by writing them back to the file, and then stop, like TEXT, back at the main menu. If the note array is empty, F8 does not update the file.

Note: Press any of the note keys, defined below, to begin sounding a note. The note ends with the next keystroke. The note keys, and the notes to which they correspond, are shown in Figure 1.

The notes range from an F in octave 1, to an F in octave 4. Shifting into upper case shifts into a higher octave range.

Figure 2



Rest: Press the space bar to begin a rest (silence). The rest ends with the next keystroke.

Enter: Press ENTER to append the sequence of notes just sounded to the bottom of the array. If no notes are waiting to be entered, ENTER plays the notes in the array, as above.

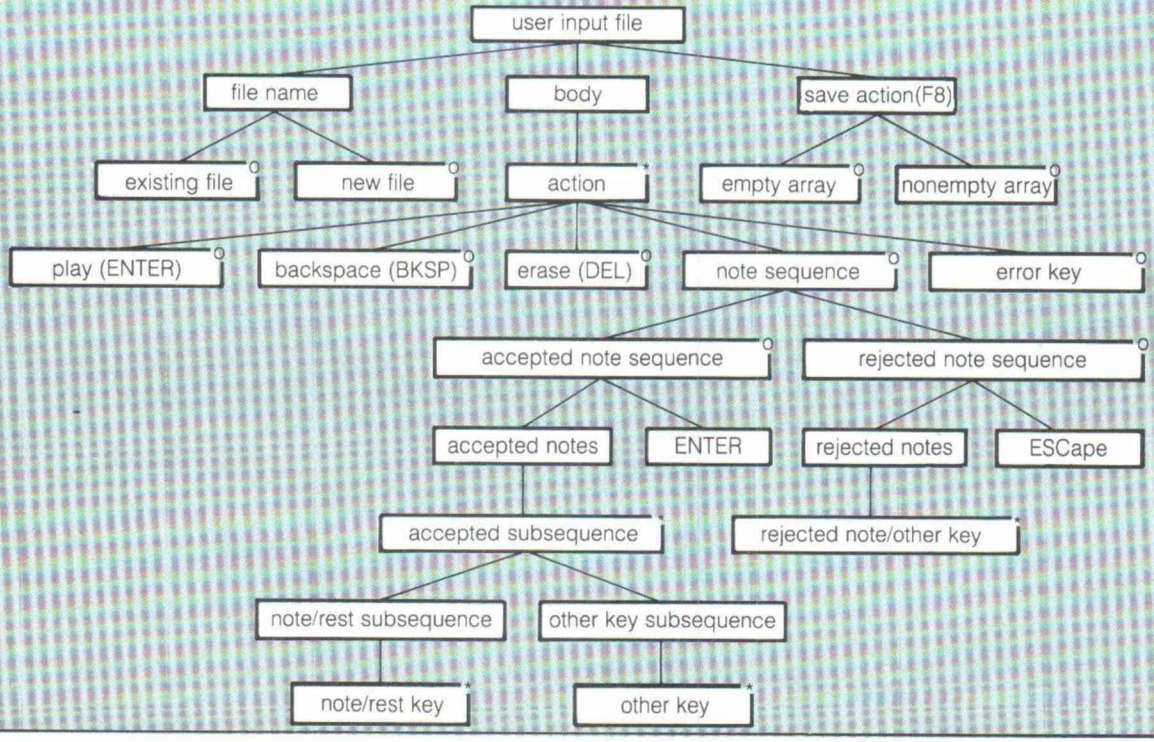
Reject: Press ESC to reject the sequence of notes just sounded, leaving the array unchanged.

This user interface is the specification for our program. Defining it is the creative part of the job. The rest of the job — designing, coding, and testing the program — is essentially a mechanical process, which can be accomplished by following a formal methodology.

DESIGNING

The program design methodology I use, Jackson Structured Programming (JSP), was developed by an Englishman, Michael Jackson. It works for most programming jobs, though it's more widely

Figure 3



used in Europe than it is in the U.S. It yields program designs that are, in a very pragmatic way, correct — in the sense that the resulting program structures contain a unique place for everything the program has to do, and for everything the program might reasonably have to be modified to do in the future! (See JSP & JSD: The Jackson Approach To Software Development by John R. Cameron, IEEE Computer Society, P.O. Box 80542, Worldway Postal Center, Los Angeles, CA 90080, for a tutorial on this methodology.)

JSP is a data-oriented methodology, so its first step is to define the data structures. The data structures our program will process are shown in the system diagram in Figure 2.

The major data structure is the sequence of user actions, which we might call the user input file, using the word "file" in a characteristically loose sense. The other data structure our program must process is the music text file, but we can properly defer considering its structure as being an implementation issue. So, we should be able to base our program structure solely on the structure of the user input file.

This structure is shown in Figure 3.

In this structure diagram, * in the upper right corner of a box indicates the element is repeated, zero or more times, while o in the upper right corner indicates the element is a selection, only one of which is chosen at any given time. Boxes without o or * indicate elements that occur in strict sequence, one after another.

Producing this data structure diagram is the hard part of the design process. We must come up with a structure that is both faithful to the data it represents, and suitable for the processing we intend.

THE CROSS CHECK

These two requirements provide a cross-check against each other, and it is typical to find out, in a later design step, that the program structure we have developed is incorrect, and we have to go back and modify the data structure on which it is based before we can proceed.

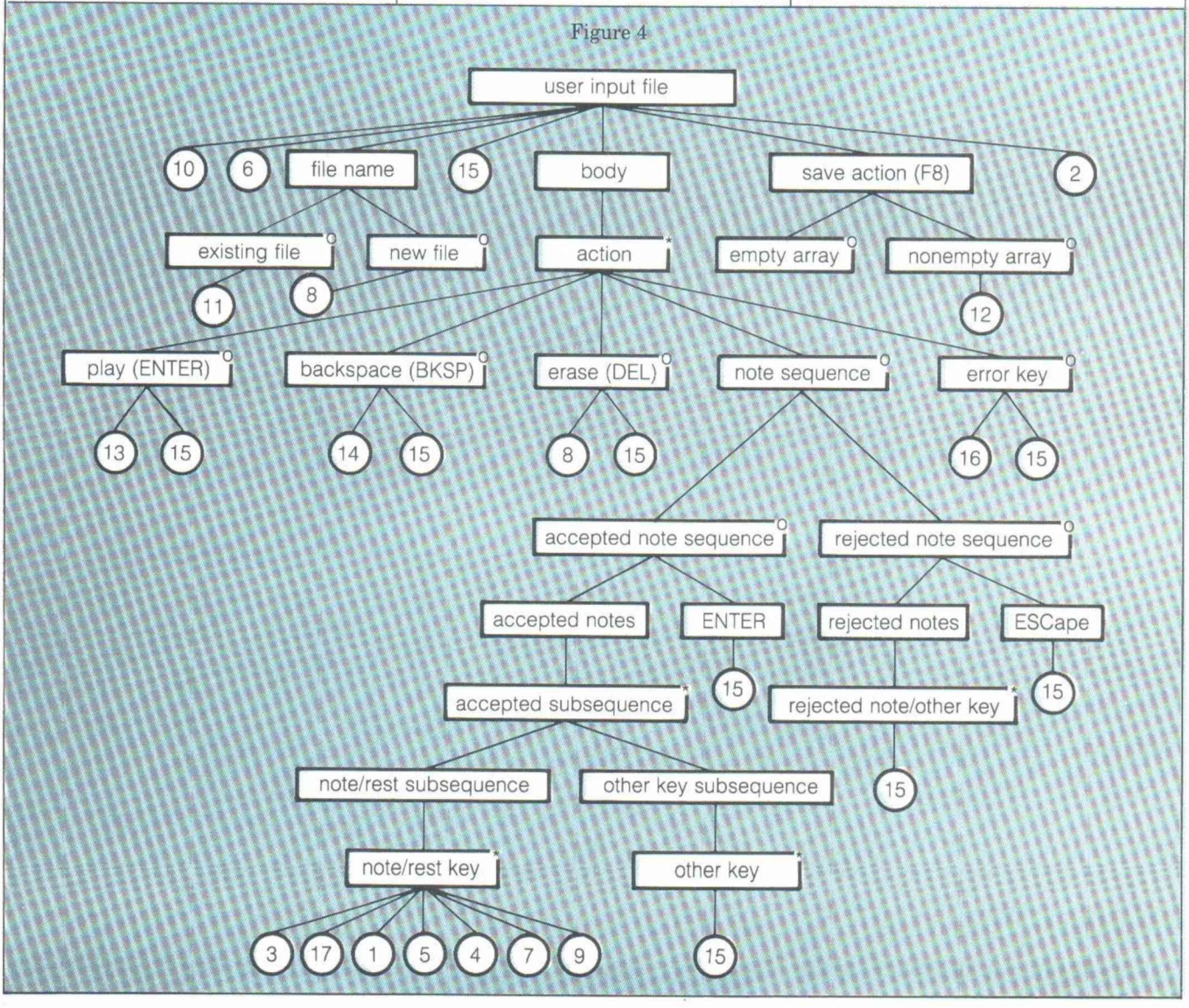
In the data structure diagrammed above, the user input file consists of a sequence of three things: the user's response to the file-name prompt; a body of user actions; and a final, save action.

The file name is either that of an existing file, or a new file.

The file body consists of a repetition of zero or more user actions.

The final, save action applies either to an empty or a nonempty array of notes.

Figure 4



MUSIC

A user action is one of the following: a play command; a backspace command; an erase command; a sequence of note-creating keystrokes; or some other keystroke, which we shall interpret as an error. This last selection is important. If we wish to design a program that can deal with any possible sequence of user keystrokes, we must incorporate all the possible errors the user can make into our data structure.

It would be a serious design mistake, at this level in our data structure, to represent "note sequence" as a selection of separate keystrokes, even though this would faithfully represent our user input file. The intermediate-level data structures we would omit by doing this will play important roles in our program, as we would eventually find out. In particular, it is always a mistake to try to ignore the different levels of repetition that occur in the data.

The user keystrokes that make up a note sequence are either accepted (with ENTER), or rejected (with ESCape), and these are the only two keystrokes that can terminate a note sequence.

An accepted note sequence is made up of a repetition of subsequence pairs: a subsequence of notes and rests, followed by a subsequence of (zero or more) other keystrokes. In other words, a series of note/rest keystrokes can be followed by a series of other keystrokes, and this pattern can be repeated.

A rejected note sequence is made up of a series of rejected keystrokes.

RULES, RULES

Ordinarily, after defining all our data structures, the next step would be to combine them, at their points of exact correspondence, into a single program structure. JSP provides rules for doing this, rules for checking the results, rules for recognizing when incompatible data structures cannot be combined, and rules for dealing with such "structure clashes."

Fortunately, we have only the one data structure to consider at this point in our design, so it's by default also our program structure, or can be made to serve this purpose by including the word "process" in each box.

So, with the program structure in hand, our next step is to list the operations that our program must perform.

Program operations can be listed in any order, and it is not very important at this stage that the list be complete. Omissions eventually become apparent, and operations can easily be added at any later stage, if our program structure is correct.

Program operations are usually described in English, and at a low enough level that their coding poses no difficulty in whatever programming language is being used. The rare exceptions to this can be handled by designing a separate subprogram to perform the operation.

It is important to realize that program operations should not include tests. Specifying the conditions that control selections and iterations is done at a later design stage. Also, the coding of an operation should not depend in any way on where in the program structure the operation occurs. This should be handled by listing a different operation for each context.

THE FIRST LIST

With all this in mind, here's a preliminary list of operations for our program:

1. Sound a note or rest, poll the keyboard, and count.
2. Return to the main menu.
3. Clear the count.
4. Convert the count to a duration.
5. Store the note in the note array at the current index.
6. Prompt for a file name.
7. Store the duration in the duration array at the current index.
8. Clear the index.
9. Increment the index by one.
10. Dimension the note and duration arrays.
11. Read the note and duration arrays from the file.
12. Write the note and duration arrays to the file.
13. Sound all the notes and rests in the note array for the durations specified in the duration array.
14. Decrement the index by one.
15. Poll the keyboard.
16. Beep for an error.
17. Convert the note key to a pitch number.

Operation 1, sounding a note, is a fundamental operation of our program. (A rest can be implemented as a note with a pitch number of zero.) Polling the keyboard, to determine when the note ends, is an integral part of it. And since the 100 keeps time in seconds, and not fractions of a second, we must count the number of polling operations to determine the duration of the note.

Operation 13, sounding all the notes and rests in the array, is the playback operation.

Except for operations 11 and 12, all of these operations are easily coded in Model 100 BASIC. Operations 11 and 12, the file read and write operations,

are difficult only because we haven't decided how we're going to store notes and durations. We can defer coding them until we get the rest of the program running. Operation 13 is a little difficult because it involves looping through the note and duration arrays, but it's simple enough not to require any further design work.

There may well be other operations we need, but they can easily be added later.

QUESTIONS

The next step in the JSP program design method is to place the operations within our program structure. This is done by asking questions of the general form, How many times do we want to do operation X? For example, How many times do we want to decrement the index by one? The answer, once per backspace action, tells us where the operation fits in our program structure. Some more examples: How many times do we want to clear the index? Once per erase action and once per new file. How many times do we want to sound a note, poll the keyboard, and count? Once per accepted note. How many times do we want to return to the menu? Once per user input

Figure 5

```
BEGIN.
| Dimension the note and duration arrays;
| Prompt for a file name;
| SELECT existing file:
| | Read the note and duration arrays from the file;
| OR ELSE new file:
| | Clear the index;
| END SELECTION.
| Poll the keyboard;
| REPEAT action:
| | SELECT play action (ENTER):
| | | Sound all the notes and rests in the note array;
| | | Poll the keyboard;
| | OR SELECT backspace action (BKSP):
| | | Decrement the index by one;
| | | Poll the keyboard;
| | OR SELECT erase action (DEL):
| | | Clear the index;
| | | Poll the keyboard;
| | OR SELECT note sequence:
| | | SELECT accepted note sequence:
| | | | REPEAT accepted subsequence:
| | | | | REPEAT note/rest key:
| | | | | | Clear the count;
| | | | | | Convert the key to a pitch number;
```

```
    | | | | | | Sound a note, poll the keyboard, and count;
    | | | | | | Store note in note array at current index;
    | | | | | | Convert the count to a duration;
    | | | | | | Store the duration in the duration array;
    | | | | | | Increment the index by one;
    | | | | | END REPETITION.
    | | | | | REPEAT other key:
    | | | | | | Poll the keyboard;
    | | | | | END REPETITION.
    | | | | END REPETITION.
    | | | | Poll the keyboard;
    | | | OR ELSE rejected note sequence:
    | | | | REPEAT rejected note/other key:
    | | | | | Poll the keyboard;
    | | | | END REPETITION.
    | | | | Poll the keyboard;
    | | | END SELECTION.
    | | OR ELSE error key:
    | | | Beep for an error;
    | | | Poll the keyboard;
    | | END SELECTION.
    | END REPETITION.
    | SELECT empty array:
    | OR ELSE nonempty array:
    | | Write the note and duration arrays to the file;
    | END SELECTION.
    | Return to the main menu;
END.
```

file. And so on.

The answers to these questions must be completely obvious. If they are not, we have a problem: Either our program structure has no place for the operation — in which case the program structure is wrong, and so, most likely, is the data structure it is based on — or we've misnamed some of the boxes in it. In either case, we've got to go back and do some rework before we can go on.

The process of allocating our 17 operations in this question and answer fashion produces the detailed program structure shown in Figure 4.

PSEUDO-CODE

Once we have allocated all our operations to their natural places in our program structures the next step is to write a pseudo-code version of the program. The JSP pseudo-code is called Schematic Logic. What it looks like is shown in Figure 5:

The next step in the JSP methodology is to incorporate, into our schematic logic pseudo-code, all the conditions needed to control the selections and repetitions. Most of the selection conditions are straightforward. For example, "SELECT backspace action" checks for

the BKSP key and a nonzero index. "SELECT note sequence" checks for a note or rest key and an index that does not exceed the dimension of the note and duration arrays. "SELECT empty array" checks for a zero index.

The repetition conditions are a little trickier. "REPEAT action," for example, checks for the F8, save action function key to terminate the main loop. "REPEAT accepted subsequence" checks for the ENTER key to terminate that loop. "REPEAT other key" checks for either the ENTER key or a note or rest key and an index that does not exceed the array dimensions to terminate the loop.

The one condition that we cannot write code to test is for "SELECT accepted note sequence." The only difference between an accepted note sequence and a rejected one is that the former ends with the ENTER key while the latter ends with the ESC key. We certainly can't code a test for a keystroke the user hasn't entered yet!

The way out of this dilemma is to assume one of the selections, to check this assumption at every opportunity, and to admit the other selection if the assumption proves false. If it does, we have the problem of backing out the operations

Figure 6

```
      BEGIN.
 100  | Dimension the note and duration arrays;
 200  | Prompt for a file name;
 300  | SELECT existing file:
 400  | | Read the note and duration arrays from the file;
      | OR ELSE new file:
 500  | | Clear the index;
      | END SELECTION.
 600  | Poll the keyboard;
 700  | REPEAT action (until F8 key):
 800  | | SELECT play action (ENTER key and nonzero index):
 900  | | | Sound all the notes and rests in the note array;
1000  | | | Poll the keyboard;
1100  | | OR SELECT backspace action (BKSP key and nonzero index):
1200  | | | Decrement the index by one;
1300  | | | Poll the keyboard;
1400  | | OR SELECT erase action (DEL key and nonzero index):
1500  | | | Clear the index;
1600  | | | Poll the keyboard;
1700  | | OR SELECT note sequence (note/rest key and nonmax index):
1800  | | | Save the index;
      | | | ASSUME accepted note sequence:
1900  | | | | REPEAT accepted subsequence (until ENTER key):
2000  | | | | | REPEAT note/rest key (until ENTER or other key)
2100  | | | | | | Clear the count;
```

```
2200 | | | | | | Convert the key to a pitch number;
2300 | | | | | | Sound a note, poll the keyboard, and count;
2400 | | | | | | QUIT if rejected note sequence (ESCape key).
2500 | | | | | | Store note in note array at current index;
2600 | | | | | | Convert the count to a duration;
2700 | | | | | | Store the duration in the duration array;
2800 | | | | | | Increment the index by one;
2900 | | | | | END REPETITION.
3000 | | | | | REPEAT other key (until note, rest, or ENTER key):
3100 | | | | | | Poll the keyboard;
3200 | | | | | | QUIT if rejected note sequence (ESCape key).
3300 | | | | | END REPETITION.
3400 | | | | END REPETITION.
3500 | | | | Poll the keyboard;
     | | | ADMIT rejected note sequence:
3600 | | | | Restore the index;
3700 | | | | Poll the keyboard;
3800 | | | END ASSUMPTION.
     | | OR ELSE error key:
3900 | | | Beep for an error;
4000 | | | Poll the keyboard;
     | | END SELECTION.
4100 | END REPETITION.
4200 | SELECT empty array (zero index):
   | OR ELSE nonempty array:
4300 | | Write the note and duration arrays to the file;
   | END SELECTION.
4400 | Return to the main menu;
   END.
```

executed during the processing of the false, assumed selection. Looking at these operations, we can ignore most of them. The only one that would cause us any trouble is the repeated execution of operation 9, increment the index. This can be easily backed out by saving the index just before we make the assumption, and restoring it when we admit the contrary. These two, additional operations are therefore easily allocated to our program structure.

UPDATED LOGIC

Updating our schematic logic to include all these conditions and considerations produces the result shown in Figure 6.

Since we are going to write Music/100 in BASIC, the numbers in the left margin will help.

The BASIC listing for Music/100 is shown below. It includes the code to read and write the specified music file.

The design of the music file read and write operations was relatively straightforward, once I decided to implement music files by storing one note per line, as a pair of numbers — a tone number, followed by a duration number — exactly as used in the SOUND statement.

Using TEXT to edit such a file, you can alter durations and tones, insert and delete notes, and copy and move note sequences.

OUR PROGRAM

The BASIC program, coded directly from the pseudo-code (Figure 6) is shown in Figure 7.

Some of the operations, notably at 300, 400, and 4300, required several lines of BASIC code. The operations at lines 110-170 were omitted from the original operations list; the need for them became apparent only during the coding stage.

The program could be optimized somewhat, to save a few bytes, but that would destroy its structure and perhaps make later enhancements more difficult.

In summary, we've seen that the first step in building any program is to determine the requirements it must meet. Some exploratory programming may be needed here to see if our computer can do the job. This is where building a prototype may come in handy.

Next, if it isn't part of the requirements, comes the user interface. Designing this is often the most creative part of the job.

Figure 7

```
100 DIM N%(200), D%(200), T%(37)
110 KEY 8,CHR$(255)
120 A$=" awsedrfgyhujkolp;['AWSEDFTGYHUJKOLP:"+CHR$(34)
130 DATA 7032,6642,6269,5918,5586,5272,4976,4697,4433,4184
140 DATA 3950,3728,3516,3321,3134,2959,2793,2636,2484,2348
150 DATA 2216,2092,1975,1864,1758,1660,1567,1479,1396,1318
160 DATA 1244,1174,1108,1046,987,932,879
170 T%(0)=0: FOR I = 1 TO 37: READ T%(I): NEXT I
200 CLS
210 LINE INPUT "File to play?  ";FN$
220 IF LEN(FN$)>6 THEN BEEP: GOTO 210
300 ON ERROR GOTO 500
310 OPEN FN$+".DO" FOR INPUT AS 1
400 X%=0: NF$="no"
410 INPUT #1,N%(X%),D%(X%)
420 X%=X%+1
430 IF EOF(1) THEN 460
440 INPUT #1,N%(X%),D%(X%)
450 GOTO 420
460 PRINT 200-X%;"notes remaining"
470 GOTO 600
500 X%=0: NF$="yes": ON ERROR GOTO 4400; RESUME 600
600 K$=INKEY$: IF K$="" THEN 600
700 IF K$=CHR$(255) THEN 4200
800 IF K$=CHR$(13) OR X%=0 THEN 1100
900 FOR I=0 TO X%-1: SOUND N%(I),D%(I): NEXT I
1000 K$=INKEY$: IF K$="" THEN 1000 ELSE 4100
1100 IF K$<>CHR$(8) OR X%=0 THEN 1400
1200 X%=X%-1
1300 K$=INKEY$: IF K$="" THEN 1300 ELSE 4100
1400 IF K$<>CHR$(127) OR X%=0 THEN 1700
1500 X%=0
```

```
1600 K$=INKEY$: IF K$="" THEN 1600 ELSE 4100
1700 P%=INSTR(A$,K$): IF P%=0 OR X%>200 THEN 3900
1800 S%=X%
1900 IF K$=CHR$(13) THEN 3500
2000 P%=INSTR(A$,K$): IF P%=0 OR X%>200 THEN 3000
2100 C%=0
2200 P%=T%(P%-1)
2300 SOUND P%,2: C%=C%+1: K$=INKEY$: IF K$="" THEN 2300
2400 IF K$=CHR$(27) THEN 3600
2500 N%(X%)=P%
2600 L%=2*C%
2700 D%(X%)=L%
2800 X%=X%+1
2900 GOTO 2000
3000 P%=INSTR(A$,K$)
3010 IF (P%<>0 AND X%<=200) OR K$=CHR$(13) THEN 3400
3100 K$=INKEY$: IF K$="" THEN 3100
3200 IF K$=CHR$(27) THEN 3600
3300 GOTO 3000
3400 GOTO 1900
3500 PRINT 200-X%;"notes remaining"
3510 K$=INKEY$: IF K$="" THEN 3510 ELSE 4100
3600 X%=S%: PRINT 200-X%;"notes remaining"
3700 K$=INKEY$: IF K$="" THEN 3700
3800 GOTO 4100
3900 BEEP
4000 K$=INKEY$: IF K$="" THEN 4000
4100 GOTO 700
4200 IF X%=0 THEN 4400
4300 IF NF$="no" THEN KILL FN$+".DO"
4310 OPEN FN$+".DO" FOR OUTPUT AS 1
4320 FOR I=0 TO X%-1
4340 PRINT #1,N%(I);",";D%(I)
4340 NEXT I
4350 CLOSE I
4400 KEY 8, "menu"+CHR$(13): MENU
```

Then, we must design the program itself, and we are well advised to base our design on the data the program must process. Trying to base it on the functions to be performed risks producing a program that cannot handle valid but obscure data cases. And it can make it difficult to add new functions.

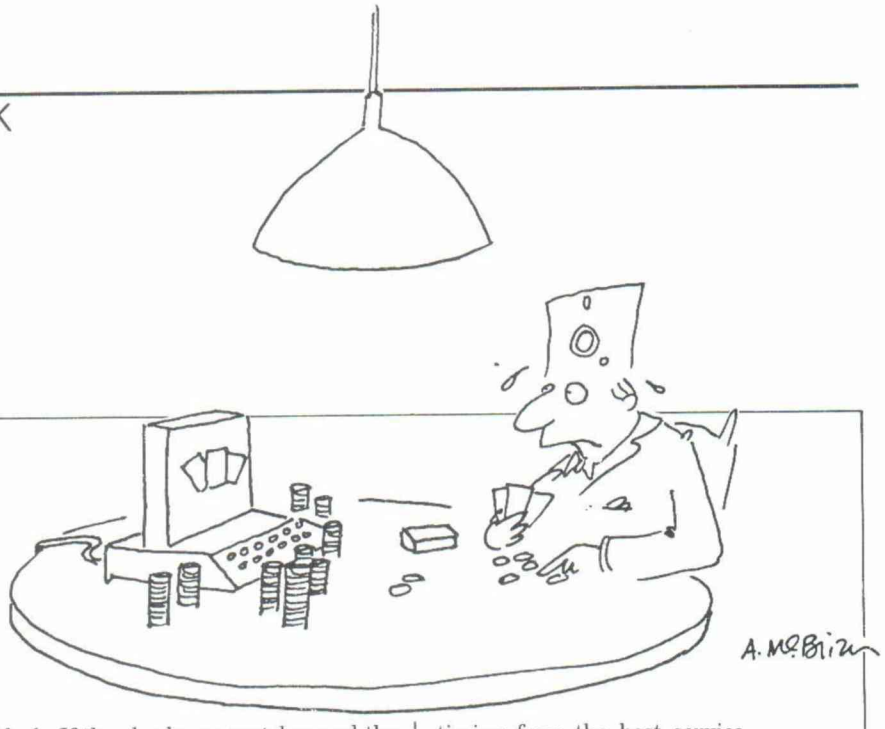
Finally, we must code and test the program. This is largely a mechanical process, and it's usually an anticlimax. Testing is always necessary, though, since operations and conditions can easily be miscoded (or omitted), and typos are always a problem.

The JSP design methodology I used in designing Music/100 is a great help in producing "correct" designs for almost any kind of program. I hope I've managed to convey some of its flavor to you, and I hope you find the program itself of value. □

This article shows how a professional computer programmer designs a musical note taker for the Model 100. This program is a musical analog of the 100's built-in TEXT program.

Please help us rate this article's value. If you've found it very valuable, circle 167 on the Reader Service Card. If it was moderately valuable, circle 168—and if it wasn't valuable to you, circle 169.

ERROR-FREE BREAKDOWN



The vast quantities of files in CompuServe's Model 100 Special Interest Group (SIG) make on-line sessions profitable (see Browsing in the Stacks, Portable 100/200, July 1985, P.9) — but downloading those files may be expensive. Why? Noise and interference may inject spurts of garbage. Or, even worse, fluctuations in telephone power levels may invisibly modify a key character or two. In a machine-language (.CO) file, this could be fatal.

CAPTURING DATA

The standard way to download text to the Model 100 or Tandy 200 is called *data capture*. Simply press F2 while in TERM mode before the transmission begins, press it again after all data has been sent, and use TEXT to edit out whatever's not needed.

If stray or erroneous characters appear in the text, nobody will know. CompuServe knows it transmitted text correctly, but the laptop has no way of checking. So, when the 353rd data byte is changed from an *F* to an *R*, you'll only find out when the computer suddenly cold-starts — and everything is lost.

XMODEM TO THE RESCUE

A new concept in data transfer is called *data packets* — the information is broken into short blocks of code, each sent separately and verified by the receiver.

XMODEM is one of the oldest and most popular of the data-packet systems. Also known as the MODEM7 and CHRISTENSEN protocol (after its creator, Ward Christensen), XMODEM ensures verifiably accurate downloading.

In a nutshell, here's how XMODEM works:

1. When the receiver is ready to accept XMODEM text, it sends a NAK signal (negative acknowledgment, ASCII 21). The NAK is sent every 10 seconds until. . .

2. The sender transmits a block of text. Each block contains the sequential block number, 128 bytes of data and a checksum. After the block is sent, the sender waits while. . .

3. The receiver attempts to verify the block. If the checksum matches and the block number is in order, the receiver transmits an ACK (positive acknowledgement, ASCII 6). If the block was received incorrectly, or if a pause of more than one second occurs during transmission, the receiver sends the NAK code. This signal is repeated every ten seconds.

4. If the sender sees the NAK signal from the receiver, it retransmits the current block. Otherwise, it sends the next sequential block.

5. Steps 3 and 4 are repeated until the end of the transmission. At the end of file, the sender transmits EOT (End of Transmission, ASCII 4) instead of the next text block.

6. When the receiver sees the EOT, it replies with ACK, ending the file transmission process.

SOME XMODEM PROBLEMS

When used with CompuServe, XMODEM often has difficulty. SIG members have reported the following problems during an XMODEM download:

- The receiver can't successfully receive even the first block of data.
- The receiver continually sends a stream of NAK signals — sometimes rejecting every block.
- The receiver suddenly aborts XMODEM in the middle of a seemingly successful file transfer.
- The receiver has difficulty and gives up between five and 20 blocks into the transfer.
- File transfer proceeds smoothly, and suddenly every block transmitted is rejected.

Often these problems may be caused by either noisy telephone lines or poor timing from the host service.

IMPERFECT IMPLEMENTATION

Why do users have difficulty with XMODEM and CompuServe, when it always works fine between two consenting microcomputers? When Christensen designed the protocol, he made a few explicit assumptions:

- File lengths are exact multiples of 128 bytes.
- Word length on both computers is eight bits, and data is transmitted with no parity and one stop bit.
- Both computers are dedicated single-user machines, and are talking directly over a telephone line.

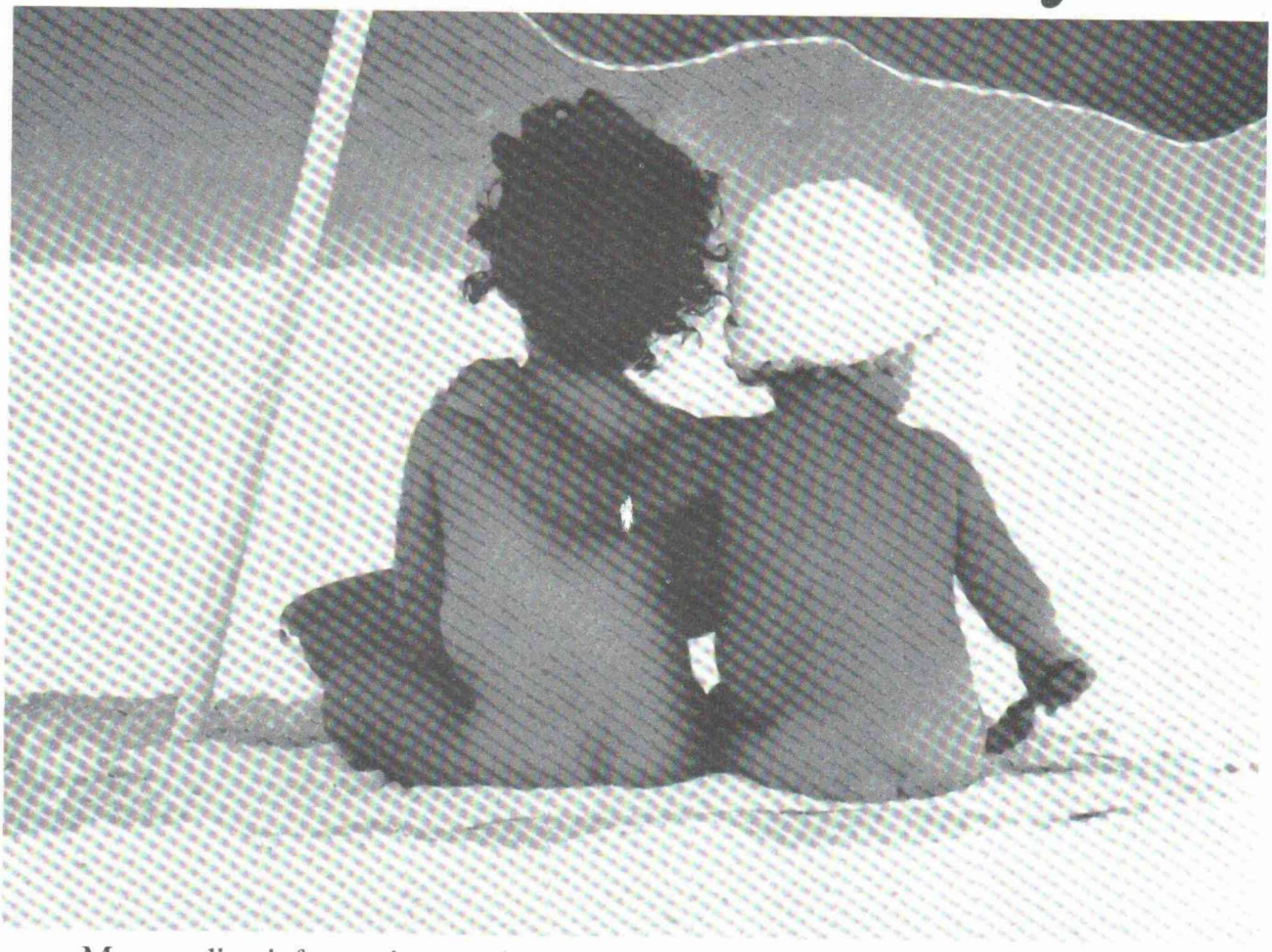
These three conditions don't always hold. In reality, files may be any length and transmission parameters vary. Also, one or both computers may be a large multi-user service working through multiple packet-switching systems. And that means large delays in transmission.

Because mainframes timeshare, and packet-switching networks like Telenet and Tymenet route so much traffic, delays of many seconds or even minutes can occur at any time. The strict XMODEM definition treats a pause of more than 10 seconds as a loss of data — and that's a short, barely noticeable pause to CompuServe.

So, when downloading from CompuServe using XMODEM, and the download fails for no apparent reason, try again. If the problem persists, try disconnecting and redialing, or waiting a half-hour to let the circuits clear. There's nothing else you can do. □

Please help us rate this article's value. If you've found it very valuable, circle 170 on the Reader Service Card. If it was moderately valuable, circle 171—and if it wasn't valuable to you, circle 172.

The Source[SM] Is Friendly.



Many online information services claim to be "user friendly." But only one really lives up to that promise.

The Source.

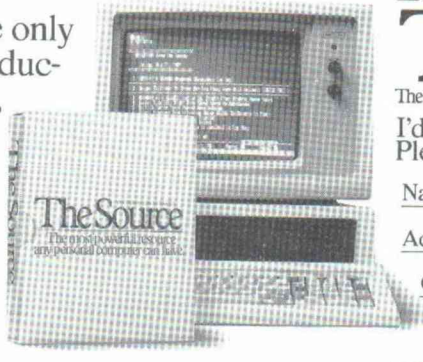
You see, The Source is specifically designed to save you time online. With new, shorter menus. Simpler commands. And a user's manual so well-written and easy to understand, it's setting an industry standard.

The Source is also the only service that gives you introductory lessons and assistance, free of on-line charges. So you get up-to-speed on our dime, not yours.

You get to the information you need in record time, without frustration. Everything from the hour's headlines to travel reservations. From special interest groups to online stock trading. So you don't waste your valuable time. Or money.

Call 1-800-336-3366, send the coupon, or visit your nearest computer dealer.

And make friends with America's friendliest online information network.



INFORMATION NETWORK

The Source[SM]

The most powerful resource any personal computer can have.

I'd like to make friends with The Source. Please send more information to:

Name

Address

City, State Zip

7826S16

Mail to: Source Telecomputing Corporation, 1616 Anderson Road, McLean, VA 22102

Circle 56 on Reader Service Card.

loop that counts up to 5,000. This 15 seconds forms the baseline for all subsequent measurements of performance.

REMARKABLE EFFECTS

The program listing labeled TEST2 shows the effects of one small change. (Remember, this program listing does not have the subroutine starting at line 500 found in TEST1. This is just to save space on the program listing. A working version of the program would require this subroutine).

The only difference between TEST1 and TEST2 is the addition of a single line with a REMark (REM). REMarks are used to document programs, and the test programs presented in this article have REMarks that identify the program and explain what's happening.

TEST2 shows that adding a single line and a blank REMark to our baseline program can have a noticeable effect. TEST2 runs in 17 seconds instead of 15 seconds. This is a slow down of 13 percent, all caused by the addition of a blank line and an empty REMark statement!

The moral of this example is not to eliminate all REMarks from your programs. REMarks are useful and sometimes necessary. But you should try to eliminate REMarks from all sections of code, such as loops, which will be run a great many times. For these sections of a program, REMarks should be put outside the loop.

One way to reduce the overhead devoted to REMarks is to use one of the popular "packer" programs that strips off REMarks and compress BASIC programs. The CompuServe Model 100 SIG has at least one of these stripper programs designed not only to compress the size of your programs but also to speed them up by eliminating REMarks. When using one of these programs, you usually keep two versions of the program: one that's fully REMarked so you can tell what's going on in the program, and a compressed version that you use.

HOW YOU DO IT IS IMPORTANT

TEST3 reinforces the importance of how you do something, and the effect this can have on a speedier program. TEST3 is the functional equivalent of TEST1, but instead of a FOR/NEXT loop this program uses an IF/THEN/ELSE control statement.

Both programs give you 5,000 iterations of a particular loop, but the control technique used in TEST3 will cause the program to take 48 seconds instead of 15 seconds to run. This dramatic difference in performance illustrates why data processing enthusiasts are often interested in topics like different sorting techniques or different methods of organizing and finding data. Something as simple as changing the sorting technique of a particular program can often improve performance dramatically.

SEEING IS SLOWING

Although we want to avoid any hardware tricks in this article, there is one characteristic of the 100/200 that you should be aware of because it has a dramatic effect on a program's performance.

Our computers use liquid-crystal display (LCD) screens that are comparatively slow to update. Program TEST4 dramatically illustrates the effect of this update slowness on performance.

The simple inclusion of a PRINT statement causes this program to take a whopping 1,723 seconds to run. This is almost 29 minutes, and the only difference between TEST4 and TEST1 (which takes 15 seconds to run) is that single PRINT statement.

The PRINT statement sends a carriage return and linefeed combination to the LCD. This causes any information which was on the screen to scroll up. Because of the time involved in updating the screen, the performance of the program is dramatically reduced.

It's usually good programming practice to inform a user what the program is doing when it gets involved in a time-consuming process like sorting a long list. Many programmers also like to provide an update of the progress of the program as it performs this operation. This example shows that while you should tell a user the program is about to start a time-consuming process (with a message such as "SORTING NOW IN PROGRESS"), you will greatly increase the speed of the program if you don't include any PRINT statements to provide updates in the middle of this process.

MAKING IT FASTER

TEST4 shows how a characteristic of a 100/200 can make your program run drastically slower, but how do you go about making it run faster?

One thing you can do is to take the hints provided in this article, as well as other articles in Portable 100/200 magazine, and apply them in a methodical fashion.

You'll soon learn most programs follow the 80-20 rule, which states that about 80 percent of the program's time will be spent executing 20 percent of the code.

If you're just hacking around for the fun of it, shaving off a few microseconds from any place in a program can be an interesting pastime, but if you're serious about being productive in your programming efforts, then you'll be better off concentrating on just those sections of code which take up most of the time during execution.

By concentrating on the code which is executed the most frequently, you'll make the most effective use of your time. This is an important issue if you're looking at program productivity from a business standpoint, instead of as a hobby. For instance, if you have a program that you run once a week and you spend eight hours improving execution speed by five minutes, you would have to run the program for almost two years before the time you saved on running the program equals the time you spent on improving the speed.

More importantly, if your program follows the same 80-20 rule most programs do, you could probably gain most of that five-minute speed-up by just concentrating on a few sections of the program, instead of spending time to optimize every line.

HOW TO DO IT

To speed up your own programs, find where the program spends most of its time. Using a time measurement subroutine such as the one presented in the test programs in this article, you can track the performance of various subroutines. All you have to do is include subroutine calls at the beginning and end of major sections of the program, printing out appropriate messages as to where the program execution is and how much time is being taken to accomplish that part of the program.

For instance, in the simple test program in this article, you could print a simple message that says "ENTERING 1 TO 5,000 TEST LOOP" and then print the total elapsed time after the loop was completed. You could do the same for other sections of more complicated programs.

If you do any sorting, large mathematical calculations, intensive screen

TURBOCHARGE

```
5   REM test1
8   CLS:PRINT "TEST1 - START AT ";TIME$
10  START$=TIME$
20  FOR I=1 TO 5000
30  NEXT I
40  FINISH$=TIME$
45  PRINT"FINISH AT ";TIME$
50  GOSUB 500
499 END
500 HR=(VAL(LEFT$(FINISH$,2))
    -VAL(LEFT$(START$,2)))*3600
505 'hour value converted to seconds
510 MIN=(VAL(MID$(FINISH$,4,2))
    -VAL(MID$(START$,4,2)))*60
515 'minutes converted to seconds
520 SEC=VAL(RIGHT$(FINISH$,2))
    -VAL(RIGHT$(START$,2))
525 'seconds okay
530 T=HR+MIN+SEC
535 'add elapsed time in seconds up
540 PRINT:PRNT"Elapsed time "T" seconds":PRINT
600 RETURN
```

```
5   REM test2
8   CLS:PRINT "TEST2 - START AT ";TIME$
10  START$=TIME$
20  FOR I=1 TO 5000
25  REM
30  NEXT I
40  FINISH$=TIME$
45  PRINT"FINISH AT ";TIME$
50  GOSUB 500
499 END
```

```
5   REM test3
8   CLS:PRINT "TEST3 - START AT ";TIME$
10  START$=TIME$
15  I=1
20  I=I+1
30  IF I=5000 THEN 40 ELSE 20
40  FINISH$=TIME$
45  PRINT"FINISH AT ";TIME$
50  GOSUB 500
499 END
```

```
5   REM test4
8   CLS:PRINT "TEST4 - START AT ";TIME$
10  START$=TIME$
20  FOR I=1 TO 5000
25  PRINT
30  NEXT I
40  FINISH$=TIME$
45  PRINT"FINISH AT ";TIME$
50  GOSUB 500
499 END
```

```
5   REM test5
8   CLS:PRINT "TEST5 - START AT ";TIME$
10  START$=TIME$
```

```
20  FOR I=1 TO 5000
30  NEXT
40  FINISH$=TIME$
45  PRINT"FINISH AT ";TIME$
50  GOSUB 500
499 END
```

```
5   REM test6
8   CLS:PRINT "TEST6 - START AT ";TIME$
10  START$=TIME$
20  FOR I%=1 TO 5000
30  NEXT
40  FINISH$=TIME$
45  PRINT"FINISH AT ";TIME$
50  GOSUB 500
499 END
```

updates or printing, these are prime candidates for measuring how much time is being spent and working on ways to increase the efficiency of these sections of code.

Once you identify where most of the time is being spent on the program, then even subtle differences can make surprising improvements in the execution speed of the program. Example TEST5 shows this.

Here, the only change is to eliminate the name of the variable I when doing the NEXT command. Model 100/200 BASIC allows you to do this.

By not having to check for the variable I, the program will speed execution from 15 seconds to 13 seconds. This is a 13 percent improvement, which is quite reasonable for such a little change.

The key is to make changes one at a time and to measure the difference. Sometimes you may find that changes don't improve the program and, in fact, make execution speed slower.

Continue improving the program in a step-by-step fashion. For instance, even the simple program we've used in this article can have a dramatic speed reduction through continued refinement. TEST6 shows this.

Using what we learned in TEST5, we've eliminated the check for the variable name at the end of the next loop. In addition, we've made the loop variable an integer (I% indicates an integer variable in Model 100/200 BASIC).

The result of these two changes is now the entire loop runs in only three seconds. This is an 80 percent reduction in the time required for the baseline case represented by TEST1 — a dramatic reduction showing that even a simple program, such as the test program presented in this article, can be greatly speeded up by attention to detail, step-by-step improvement, and by concentrating on those sections of the program where most time is being spent.

THE STEP-BY-STEP APPROACH

In this article we've tried to present you with a series of hints for speeding up your BASIC programs. While hints are valuable, a more important process is to understand how you can methodically apply the techniques shown to speed up your own programs.

- Remember the 80-20 rule and concentrate your efforts on sections of the program where most time is being spent. This makes most efficient use of your time. Use a timing subroutine like the one presented in the example programs to discover where the program spends most of its time.
- Once you've isolated where the program is spending most of its time, pay attention to how you're doing things in the program. This means things like sorting techniques, how you're calculating values and other time-consuming tasks. It also means using appropriate commands, such as a FOR/NEXT loop instead of IF/THEN/ELSE tests.
- Avoid time-consuming tasks like unnecessary screen updates, the inclusion of REMarks within loops and other inefficiencies that slow the program down.
- Pay attention to detail. Once you've isolated sections of code you want to concentrate on, changes in details can have major results. Even simple things like changing loop counters to integers can cause a dramatic speed-up in execution.

These techniques don't have to be burdensome, but they should be developed as part of your programming habits as you work with your Model 100/200.

The results can change your programs from the BASIC equivalent of a meek-and-mild Nissan Sentra to a programmer's fire-breathing Nissan 300 Turbo ZX! □

Please help us rate this article's value. If you've found it very valuable, circle 173 on the Reader Service Card. If it was moderately valuable, circle 174—and if it wasn't valuable to you, circle 175.

ULTIMATE ROM (from page 35)

pany to market ROM-based applications. Polar Engineering and Consulting pioneered the process, and Portable Computer Support Group markets a variety of plug-in programs. But Traveling Software is the first company to package three full-scale programs on one 32 kilobyte (K) chip. Considering that the cassette versions of the three programs comprise something over 33K, that's quite an accomplishment. But we're getting ahead of ourselves.

THE ULTIMATE ANSWER

The Ultimate ROM is a plug-in cartridge that holds three of Traveling Software's most popular cassette-based programs: Idea, T-base and T-Writer. The three-in-one packaging makes for a more attractive purchase price.

Idea is an outline processor that was favorably reviewed in beta-test form in *Portable 100/200* (March 1985). The cassette-based version reviewed then was slow —it couldn't keep up with even the slowest typist. It still can't, but it is nonetheless a useful program. It allows the user to set up a hierarchy of headings, subheadings, sub-subheadings and so on to keep track of concepts, goals or procedures. Function keys let you add subheadings, rearrange items, delete entries and generally manipulate the outline.

T-base was reviewed, quite favorably, in the October 1984 issue of Portable 100. Rather than repeat what was said there, let me merely summarize: T-base is a relational database program, allowing items in one database to point to (relate to) items in another. It is as powerful as any software available for the Model 100, and likely the best database manager.

In the cassette-based format reviewed earlier, the program resided in RAM, limiting the maximum size of the data file. With T-base in ROM, almost all of RAM can be used for the database. T-base is the hardest of the three programs to learn to use, but once you get comfortable with it, it serves you well.

T-Writer is a nice text formatter with mail-merge capability. Simple dot commands (lines starting with a period) establish and modify margins, pagination and so on. It also handles automatic renumbering of paragraphs, headings, etc. It was reviewed in the September 1985 issue of *Portable 100/200*.

SO HOW'S IT WORK?

Installing the Ultimate ROM involves hardware (plugging in the ROM) and software (CALLing a subroutine in BASIC). Next thing you know there are five new names (four with the curious extension of .TS) on the menu. Fortunately the five new files only consume a few bytes of RAM, so there is little penalty in leaving them in the directory.

The ROM chip is keyed to fit slots in the option socket, so it's impossible to install it incorrectly. The flexible printed-circuit (PC) board surrounding the chip seems a bit fragile; it might get worn out after a few dozen removals and replacements.

If and when the chip is to be removed, there is none of the usual worry about having to reset RAM hooks. Instead, removing the software is accomplished by simply killing the five files which appeared when the ROM was installed.

The box containing Ultimate ROM, at over two pounds, is easily the heaviest software package on the market today for a laptop. Since the chip itself is negligible in size and weight, virtually all this mass is documentation.

(continued on page 66)

THE POLAR CONNECTION

The widespread popularity of laptop computers is due largely to the fact that the operating system and application programs are in read-only memory (ROM) rather than magnetic media such as disk or tape. Disk and tape drives are bulky, heavy and power hungry. Use of ROMs enables designers to trim weight to a minimum and use mere AA batteries for power.

Software developers like ROM software because piracy is curtailed. As a result, the purchase price can be lower than for comparable disk-based programs. Computer users like ROM-based software because there's so little to go wrong — turn on the power and there's the software with no need to fumble around for the right disk. There's no need to worry about accidentally putting a disk near a magnet and erasing it.

The Model 100 and its Kyocera brethren, the Tandy 200 and NEC PC-8201, each have an expansion ROM socket. The address lines provided there allow addressing up to 32K of memory. The idea is that any software provided need not consume valuable RAM file-storage area.

NON-STANDARD SOCKETS

Unfortunately for third-party providers such as Portable Computer Support Group (PCSG) and Polar Engineering and Consulting (PEAC), the physical arrangement of the 28 pins at the ROM socket does not match the electrical requirements of any commercially available erasable programmable read-only memory (EPROM) chip. PCSG uses an epoxy printed-circuit board to swap pin connections as necessary to allow normal EPROMs to be plugged in. PEAC uses a very thin, flexible printed-circuit board.

The first option ROMs were 8K EPROMs with two address lines unattached. As a result, the 8K of information appeared four times in memory, starting once each at 0000H, 2000H, 4000H and 6000H. Now the bugs have been worked out for full 32K ROM devices to plug into option ROM sockets.

It takes more than hardware savvy to get full use of the option ROM capability — the "trap doors" left by Microsoft in the operating system software have been uncovered and exploited. The great software achievement demonstrated in the Ultimate ROM is that BASIC programs heretofore executed in RAM may be burned into ROM and executed there. The magnitude of this accomplishment becomes apparent when you realize the option ROM and the system ROM sit in the same address space — when one is selected the other is deselected.

THE POLAR ALLIANCE

Traveling Software gained its firmware expertise via an alliance with PEAC president Tom Bennett. The agreement allows Traveling Software to use Polar's firmware technology.

This BASIC-in-ROM technology is also available to the public under the name Guardian.

—Carl Oppedahl

ULTIMATE ROM (from page 64)

Each of the three major components (Idea, T-base and T-Writer) has a manual with well over 100 pages. These are precisely the same manuals provided with the cassette-based versions of the programs. Addenda spell out changes that must be made due to the ROM medium.

Each manual is attractively bound and typeset, and well-oriented to the new user. Regrettably, only the Idea manual has an index.

Quick reference cards are provided for T-base and T-Writer. A memory management program and sample files for all three programs are provided on cassette. The reverse side contains an audio introduction by the folksy Traveling Professor — recorded, apparently, during his pre-sabbatical days.

The documentation urges you to submit problems by mail rather than phone. I found that phone calls were nonetheless answered politely and, often as not, yielded answers to my many questions. Version numbers are provided with the software which facilitate troubleshooting in the event a customer has a different version than the customer service representative.

In addition to the mail-in and dial-up support, Traveling Software offers a 30-day money-back guarantee.

BUT WHAT ABOUT FLEXIBILITY?

The ROM-based versions of Traveling Software's best-sellers function identically in most respects to their cassette-based predecessors. The few changes are actually improvements — ROM-based applications simply work better than programs you have to load into RAM.

Traveling Software devised a creative solution to one of ROMware's greatest potential drawbacks: its immutability. Many Idea, T-Writer and T-base users modify the programs to match peculiarities of their printers or to conform to special requirements of their particular applications. This is a simple matter with RAM-resident programs; the user simply LOADs the program and invokes the editor. Abundant examples are provided in the programs' manuals.

Since ROMware can't be modified, these users are left out in the cold, right? Wrong. Traveling Software allows users to specify performance characteristics through use of a configuration file called CONFIG.DO.

For instance, T-Writer includes T-Merge, which allows the operator to send a boilerplate letter to a mailing list set up in a file called ADRS.DO. The CONFIG.DO file lets the user specify a different default name for the address

file. Similarly, the operator could use asterisk commands instead of dot commands in T-Writer by changing one line in the CONFIG.DO file. The CONFIG.DO file allows users to send output to the LCD or to a serial printer instead of the default parallel printer. It also lets DVI users redirect output to a 25-line CRT.

It's a creative solution to the immutability problem — and represents a thoughtful approach to software design.

DESKTOP LINK

Traveling Software's Mark Eppley likens the Ultimate ROM-equipped laptop to a portable workstation loaded with ThinkTank, dBASE II and WordStar. He's the president of the company; it's easy to attribute this overstatement to enthusiasm.

But the claim is backed up by a file-conversion program currently in beta test. The program, which runs on desktop computers that operate under MS-DOS, converts ThinkTank files to an Idea-readable format. The files may be downloaded to the laptop via direct connection or modem, then used on the road.

The Ultimate ROM retails for $229.85 — the same figure you get if you add up the suggested retail prices of the cassette versions of Idea, T-base and T-Writer. (A $199.95 introductory price expired September 20). Given the performance improvements ROM implementation offers over the already powerful original programs, it's a bargain.

But the company doesn't stop there. Current Traveling Software users can mail in their original Idea, T-base and T-Writer program cassettes and receive $10 per cassette off the price of the Ultimate ROM. "It's our way of supporting the current users," Eppley says. "After all, they supported us."

The Ultimate ROM is memory-efficient, cost-effective, trouble-free and powerful. The documentation is adequate and the performance is bug-free. If you're in the market for an idea processor, a database manager or a text formatter, don't think twice: We recommend it. □

Carl Oppedahl, a lawyer in Manhattan specializing in technological litigation, is author of Inside the TRS-80 Model 100, Weber Systems, Chesterland, Ohio.

Please help us rate this article's value. If you've found it very valuable, circle 176 on the Reader Service Card. If it was moderately valuable, circle 177—and if it wasn't valuable to you, circle 178.

ANNOUNCING!!

100 MARKETPLACE 200

A New Classified Ad Section to Serve 100/200 Users

Got some hardware/software you want to sell or trade? Got a BBS or a User Group function to promote?

If so, then a classified listing in **MARKETPLACE** is for you.

Here's all you have to do:

1. Complete the coupon below (legibly, please!).
2. Include form of payment.
3. Mail to the address below.

Each number series (telephone, street, zip code, price, catalog, memory) counts as one word.

Minimum number of words is 18 (approx. 3 lines).

MULTIPLE INSERTION DISCOUNT: You may subtract 10% for each insertion after the first one.

DEADLINES: The day your ad and payment must be in our offices is on the 15th day of the second month preceding the cover month (in order to be in the August issue, copy and payment must be received no later than May 15).

MARKETPLACE Form

Please use this form to write your classified ad. Each space is one character. Please leave one space between words, numbers, abbreviations. If necessary, please start a new line rather than hyphenating a word. Make sure all your information is correct. Please count the words accurately, fill in the amount enclosed and credit card information or check the payment enclosed box. (Even if you place your order by phone, this form may help you organize your ad message and figure costs). Place in envelope, and mail today.

(This grid, if filled, is equivalent to a one-inch box. If you need more space, please use a separate sheet of paper. When filling the grid, please leave one open space between words.)

Number of words _____ × $2.00/word = $_____

☐ One Inch Box ($72) ☐ Screen Tint, add $10. Number of Insertions?_____

Name: _____

Address: _____

_____

Phone: (_____)

☐ Visa/MasterCard # _____ Exp. Date_____

☐ Payment Enclosed Amount $_____

Mail to: Portable 100/200 **Marketplace**
P.O. Box 250, Highland Mill, Camden, ME 04843

Telephone: 207-236-4365

Wanted

WANTED: Any information (especially printed) that would help determine if a program written for the model 100 will run on the NECPC-8210A. Also wanted - info on specific differences between 100 and NEC. Conversion hints helpful. Write: A. Johnson, 108 Camp St., Middletown, CT 06451. Thanks!

Hardware

FOR SALE — M100 (32K) plus printer and telephone cables, lucid and disk + rom's, barcode reader, ccr82, acoustic coupler, chip-tote and Radio Shack briefcase, full library of business software. $599 obo. Werner Hoermann, 600E Hermosa Dr., San Gabriel, CA 91775.

FOR SALE — PG Design 32K RAM module, works perfectly, with documentation, $180. R. Whitehead, 1 Quorn Hunt Road, West Simsbury, CT 06092.

Carrying cases. Aluminum and composit types. Foam and/or partition interiors. Free catalog. Call or write IPC, 243 San Pedro NE, Albuquerque, NM 87108. (505) 256-3561.

Instruction

PIANO — Amuse your friends, educate the kids with keyboard sound/graphics. $15. Tom Wade, 6030 Fennell, San Diego, CA 92114.

M100 OWNERS MANUALS — 150 copies free to a qualifing educational institution. Auto-wares, Inc., 440 Kirktland S.W., Grand Rapids, MI 49507. 616-243-2125

MODEL-100 QUICK REFERENCE CARD. Instructions for all programs. No more flipping through your manual! Only $2.50. Eighty Computing, P.O. Box -154, Orinda, California 94563.

"Computer Writer." The Newsletter for writers who use computers. Dynamite. $3 sample. $15 yr. DENNIS, 1332 Manitoba -1, S. Milwaukee, WI 53172.

Services

Portable Computer Message System (PCMS) — Dial: (914) 693-0293, 8 Bit, No Parity, Full Duplex, 300 Baud.

Software

MAGIC FOR YOUR 100. Knock your friend's socks off!! Six magic programs written by a professional magician. Includes "Impuls": sure to fool top programers, as well as magicians. Some practice required. $19.95 — Ned Way, PO Box 15182, Orlando, FL. 32858.

MON100 machine language monitor 16K $30; LOK100 password security 8K $19; Morse Code tutor 16K $19; Basic screen dump 8K $19; Count-Write fast word and sentence tally 8K $19; Home Inventory 24K $19. Structured Software Services 9233 NE 269th St., Battle Ground WA 98604.

Pick football winners by computer using my basic formula and game statistics. $10. Howard Allen, Box 66, Middleburg, VA 22117.

NEC-PC8201 Software. Many Categories. Catalog $1.00 refundable w/order. Chris Reid, MCSC Software, 8486 Downs Road, Winston, GA 30187, 404-489-2119.

Perfect tool for the traveling salesperson, "CALL-BACK" helps you manage phone calls. Return calls by time zone (based on area code) and date. Reschedule missed calls. Log call-back attempts, elapsed time, and memo information. For 100/200. $39.95 for cassette & manual. Requires 16K machine. SOF/SUS, INC., 4306 Upton Ave., Minneapolis, MN 55410. (612) 929-7104. MC/VISA.

User Groups

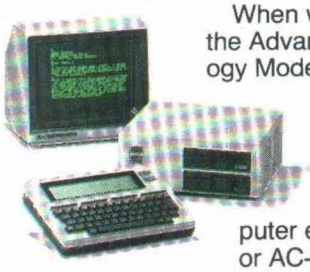
Custom medical software for 100/200. Clinical & business programs. Priced from $39.95. Richard Elliott, MD, 42 Walnut Lane, Davis, CA 95616.

ADVERTISERS INDEX

PG#	RS#	ADVERTISER
66	1	A & J Micro Drive
69	5	ARMS
32	2	Acroatix
27	3	Axonix
54	4	Buddy System
67	7	CISS
62	8	CSA Software
46	6	Cabbage Cases
CIII	10	CompuServe
60	9	Coriolis
55	11	Covington Group
50	12	Covington Group
56	13	Cryptronics
7	14	Custom Software
61	15	DFW Computer Center
46	16	Delker Electronics
61	30	Dresselhaus Computer Products
4	23	Economy Computers
69	19	Elexor
64	21	Encompass
63	22	FACS
65	20	Fort Worth Computers
64	24	Kangaroo Video Products, Inc.
65	18	Life Sciences Software
66	25	Loquor
68	—	Marketplace Ad/Section
4	27	Merritt
23	26	Micro Demon
52	28	Micro Peripheral
4	29	Nictrix
52	31	Node Computer Systems
12	33	P.C.S.G.
16	37	P.C.S.G.
29	35	P.C.S.G.
21	36	P.C.S.G.
1	32	P.C.S.G.
41	34	P.C.S.G.
CII	39	PG Design
3	38	PG Design
57	40	PIC
53	41	Paramount Marketing, Inc.
10	44	Polar
24	44	Polar
50	53	Pomerleau Computing Systems
4	42	Portable Basics
51	—	Portable Program Review
18	17	Prairie Power
9	49	Purple Computing
67	46	Purple Computing
15	45	Purple Computing
47	43	Queue Software Systems
50	55	RMD & Associates
37	47	Radio Shack
71	48	Radio Shack
53	50	Sigea
4	51	Sigea
45	52	Softmate
60	54	Southworth
59	56	The Source
38	58	Traveling Software
CIV	57	Traveling Software

THE TANDY 200 AND MODEL
OFFER MUCH MORE THAN J

Our accessories make them the only co



When we introduced the Advanced Technology Model 100, it quickly became the most highly-acclaimed lap-size computer ever. A battery or AC-powered computer with five built-in programs and a telephone modem that fits into a briefcase. Our Tandy 200 took this great idea a few steps further—a big easy-to-read 16-line by 40-character "pop-up" display, Multiplan spreadsheet software, plus an alarm and calculator.

But the story doesn't end there—we back both machines with extensive accessories that help make them powerful desktop tools.

New Low Price! Disk/Video Interface

Turn your Tandy 200 or Model 100 into a versatile disk-based home or office system with a Disk/Video Interface—now only $499 (26-3806, was $799.00 in Cat. RSC-14). It features a built-in 184K floppy disk drive, and a 40-character by 25-line display with your TV or an 80 x 25 display with a standard video monitor. You can even add a second internal disk drive.

Thermal Ribbon Printer

Use it anywhere! The TRP-100 (26-1275, $299.95) works on batteries or AC power (adapter included), and prints on plain or thermal paper. The TRP-100 quietly prints 50 characters per second. Produces bit-image graphics, too.

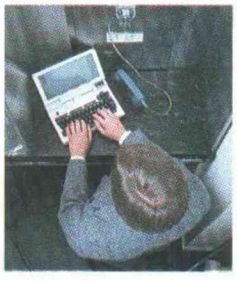
Versatile Bar Code Reader

Our Bar Code Reader (26-1183, $99.95) is an optical scanning device that's ideal for inventory control, billing and couponing. Software drivers, callable from BASIC, allow you to read Universal Product Code, 3 of 9, Plessey and other bar code formats.

100 PORTABLE COMPUTERS
ST COMPACT PORTABILITY

puter you'll need—in or out of the office.

Communications Power



One of the most exciting features of the Tandy 200 and Model 100 is the built-in modem that lets you communicate with other computers over the phone line. Our Direct-Connect Modem Cable comes with one free hour on both the CompuServe Information Network and Dow Jones News/Retrieval service.

Get news, weather and sports, business and stock market information and more. Information networks open up a whole new world of applications power.

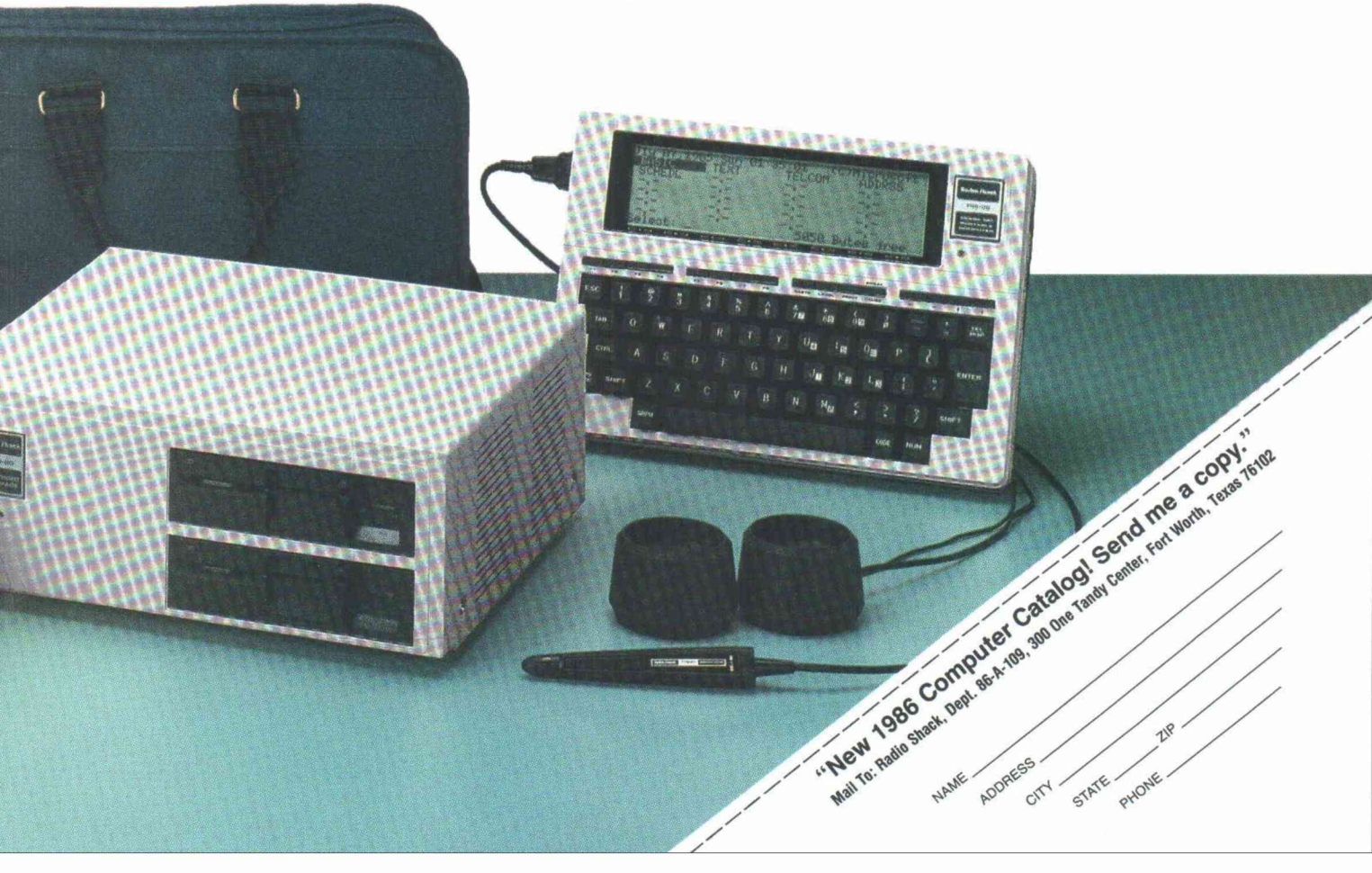
And when you're on the road, you can bypass direct connection with our Acoustic Coupler. It's a great way to keep in touch with your home office. And we even offer a briefcase and padded carrying case to make portability all it's supposed to be.

Come in Today!

Stop by your local Radio Shack Computer Center and see both the Advanced Technology 24K Model 100 (26-3802, $599) and 24K Tandy 200 (26-3860, $999), as well as our complete line of accessories and software. You'll find everything you need for professional, portable computing power—for your desk or on the road.

Radio Shack®
The Technology Store™
A DIVISION OF TANDY CORPORATION

Prices apply at Radio Shack Computer Centers and participating stores and dealers. Multiplan/TM Microsoft Corp. CompuServe/Registered TM CompuServe, Inc. Dow Jones News/Retrieval/Registered TM Dow Jones & Co., Inc.



"New 1986 Computer Catalog! Send me a copy."
Mail To: Radio Shack, Dept. 86-A-109, 300 One Tandy Center, Fort Worth, Texas 76102

NAME _____
ADDRESS _____
CITY _____ STATE _____ ZIP _____
PHONE _____

PRINTER COMPATIBILITY

Epson's P-80 printer was designed to be used with the Geneva PX-8 portable computer. However, it *can* be used with other portable computers such as the Radio Shack Model 100 and NEC's PC-8201A.

At first blush, the P-80 and Model 100 have all the signs of being incompatible. The P-80 is a serial printer, the Model 100 has a parallel printer port. The P-80's printer port uses a six pin plug while the 100's serial port has a 25 pin connector.

The solution is to build a cable specially configured to connect the two devices.

With the Model 100 data can be sent to the P-80 via its serial port. Scripsit 100, the word processing program, suppports as serial printer. Here are the pinouts to connect the P-80 to the Model 100:

P-80	Model 100
pin 1	not connected
pin 2	pin 6
pin 3	pin 2
pin 4	not connected
pin 5	pin 7
pin 6	not connected
pin 7	pin 7

You can output files from a BASIC program using OPEN and COM commands. Here's an example showing the exact syntax (it will output the word TEST to the P-80):

```
10 OPEN "COM:38N2D" FOR OUTPUT AS #1
20 PRINT #1, "TEST"
```

For the NEC PC-8201A, the cable configuration is slightly different. Because of handshaking problems, speed is limited to 300 baud.

The cable must be constructed in the following manner:

P-80	PC-8201A
pin 1	pin 1
pin 2	pin 20
pin 3	pin 2
pin 4	pin 4
pin 5	pin 7
pin 6	pin 6
pin 7	not connected

The following BASIC program outputs to the P-80:

```
10 MAXFILES=3
20 OPEN "COM:3N82NN" FOR OUTPUT AS#2
30 PRINT#2, "TEST"
```

Portable Meter Maid

As we all know, portable computers have made it into a number of markets. Publishing, finance, real estate and life insurance to name but a few. Now it seems, they are making their way into lesser known areas such as parking control.

That's right, the next parking ticket you get could be a computer printout. Cardinal Computers, Inc. of Carrollton, Texas has developed what they call TickeTrak, a computerized citation system.

TickeTrak, developed on the Epson HX-20, improves data handling of parking violations. Choice of the HX-20 was primarily due to the built-in printer.

In practice, handwritten citation books are replaced by portable battery powered computers. TickeTrak prompts the ticketing officer for all violation details with easy to use multiple choice questions. Once the data is entered, the computer prints a "ticket" to place on the vehicle and stores the data on cassette. At the end of the day, the data is transferred from the tape to a larger computer.

The benefit of such a system is not just improved control of ticket data but increased speed and accuracy in processing the data — no more deciphering illegible handwriting. Because of increased efficiency, it saves money as well.

TickeTrak received a lot of publicity when Southern Methodist University was awarded the National Association of College and University Business Officers (NACUBO) cost-cutting award for saving the campus police department more than $21,000 in violation-related costs. Several other universities and municipalities have expressed an interest in TickeTrak.

In addition, the system has virtually eliminated driver hostility. "Usually, you don't get any positive reponse from tickets, no matter how polite the police are," says Bill Caffee, SMU's police chief. "People are more interested in *watching* police write a citation now." □

Portable XMODEM

Until recently, most XMODEM implementations were on desktop microcomputers. Software like Telecommuter from Sigea Systems, Lync from Norton-Lambert and Crosstalk from MicroStuf contain XMODEM data-packet ability.

Some clever programmers have adopted the error-free protocol to the Tandy laptops, however. Here are a few of the CompuServe SIG files concerning XMODEM:

Data library 1, Telecommunications:

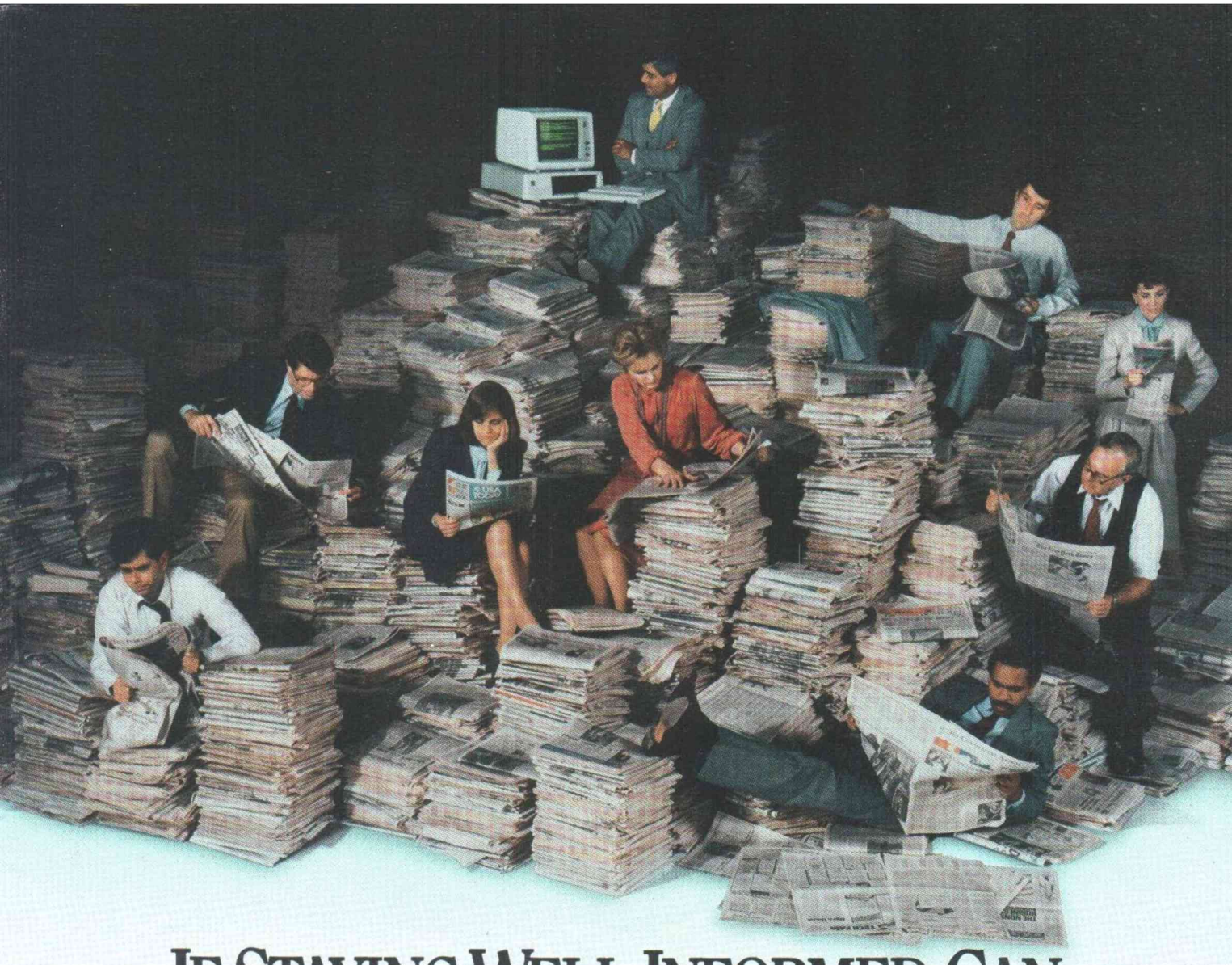
- XMODEM.312 — XMODEM for the Model 100.
- XMODEM.DOC — The documentation for XMODEM.312.
- XMODEM.200 — XMODEM for the Tandy 200.
- XMODEM.201 — The documentation for XMODEM.200.

In Data Library 7, NEC and Olivetti:

- XMODEM.DOC — The documentation for XMODEM.NEC.
- XMODEM.NEC — XMODEM for the NEC PC-8201.

There are other versions of XMODEM in DL1 that are customized for popular modems. To see the names and descriptions of the files:

Function: *dl1*
DL1: *s /key:xmodem /des* □



IF STAYING WELL INFORMED CAN KEEP YOU WELL AHEAD, COMPUSERVE'S GOT BIG NEWS FOR YOU.

Introducing the Executive News Service and state of the art access to all the news you need.

CompuServe's exclusive Executive News Service is an electronic clipping service that puts the power and scope of the entire Associated Press news wire at your fingertips. Simply enter words or phrases that interest you ... from business to politics to hometown news. The Executive News Service then screens all state and national AP wires around the clock and files selected stories for reading at your convenience.

Then there's USA TODAY Update, with late breaking business information available for quick analysis by busy executives. USA TODAY's editors monitor hundreds of national and international news sources to bring you the latest, most important developments. General news reports are updated hourly. Reports for specific industries are provided daily in a concise, executive summary format.

Or access The Washington Post, with its fingers on the pulse of the national political scene. CompuServe delivers it all, plus a variety of other information, entertainment and communications options.

You can access CompuServe with almost any computer and modem, terminal or communicating word processor. To buy a Subscription Kit, see your nearest computer dealer. To receive our informative brochure, or to order direct, call **800-848-8199** (In Ohio, call 614-457-0802). Or send this coupon today.

YES. Please send me your CompuServe Information Services brochure.

Name _____

Address _____

City _____

State _____ Zip _____

Business Phone _____

CompuServe®

Information Services
P.O. Box 20212, 5000 Arlington Centre Blvd.
Columbus, Ohio 43220
800-848-8199
In Ohio, call 614-457-0802 **ENS-1810**

An H&R Block Company

Traveling Software
introduces everything you would ever want on a single plug-in ROM chip . . .

THE ULTIMATE ROM

The ULTIMATE ROM contains three acclaimed programs:

IDEA! — One of the hottest-selling programs for the Model 100 and NEC PC-8201! Organizes anything from simple lists to complex outlines. Easily reorganizes and searches. Offers many word processing and database capabilities. Develops outlines from rough drafts into finished documents.
Original cassette price: $79.95

T-base — The database reviewers have rated as number one for the Model 100! Lets you design relational databases and set up screen files. Borrows information from fields in other databases. Performs math computations. Creates custom-printed or LCD display reports. **Original cassette price: $99.95**

T-writer — Still the favorite text formatter with owners of the Model 100 and NEC PC-8201. Prints documents created using the built-in TEXT program. Includes justification, headings, footings, underlining, italics, boldface, and more. Produces form letters and mailing labels. Word-Star-like command.
Original cassette price: $49.95

ADDED MEMORY The ULTIMATE ROM adds another 32K to your computer's memory by storing programs outside precious RAM memory space, freeing RAM for larger data files.

If your computer has a 32K memory already, for example, you can create files up to a whopping 25K in size using T-base, IDEA!, or your computer's TEXT program . . . over five hundred names and addresses in a T-base file.

ADDED CONVENIENCE With the ULTIMATE ROM you don't need to load programs individually, either. Simply plug in the chip and you're always ready to work — without using slow, clumsy cassette recorders.

The ULTIMATE ROM also works with other memory expansions, like PG Design's 64K RAM module or Purple Computing's NEC SideCar. The ULTIMATE ROM works in any active memory bank on your computer — no need to copy programs from bank to bank.

A COMPLETE PACKAGE Each of the three ULTIMATE ROM programs is complete and full-functioned — no features have been trimmed. And you have complete documentation for each program, along with an audio introduction by the renowned Traveling Professor.

THE BOTTOM LINE: IT'S A GREAT DEAL Three programs. One chip. If you order now, you can receive our free 8K "Ultimate RAM" expansion or a $25 purchase credit. If you need five or more Ultimate ROM'S, call us about our volume price discounts. Traveling Software is so sure you will like the ULTIMATE ROM, it is offering a 30-day money-back guarantee.

Only $229.85

FREE RAM OFFER
If you purchase an ULTIMATE ROM before November 30, 1985, you will get your choice of a free 8K "Ultimate RAM" memory expansion for the Tandy Model 100 or NEC 8201, or a $25 credit towards your next TSI purchase over $35.

To order call toll free 1-800-343-8080
For technical information call (206) 367-8090

Now available through Radio Shack stores by express order:

Product Numbers:
M100 90-0409
M200 90-0410

TRAVELING SOFTWARE®